ATX
MIB REFERENCE GUIDE

9031872-01

CABLETRON SYSTEMS
The Complete Networking Solution™

NOTICE

Cabletron Systems reserves the right to make changes in specifications and other information contained in this document without prior notice. The reader should in all cases consult Cabletron Systems to determine whether any such changes have been made.

The hardware, firmware, or software described in this manual is subject to change without notice.

IN NO EVENT SHALL CABLETRON SYSTEMS BE LIABLE FOR ANY INCIDENTAL, INDIRECT, SPECIAL, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING BUT NOT LIMITED TO LOST PROFITS) ARISING OUT OF OR RELATED TO THIS MANUAL OR THE INFORMATION CONTAINED IN IT, EVEN IF CABLETRON SYSTEMS HAS BEEN ADVISED OF, KNOWN, OR SHOULD HAVE KNOWN, THE POSSIBILITY OF SUCH DAMAGES.

© Copyright 1996 by Cabletron Systems, Inc., P.O. Box 5005, Rochester, NH 03866-5005
All Rights Reserved
Printed in the United States of America

Order Number: 9031872-01 May 1996

All product names mentioned in this manual may be trademarks or registered trademarks of their respective companies.

Printed on Recycled Paper

FCC NOTICE

This device complies with Part 15 of the FCC rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

NOTE: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to Part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment uses, generates, and can radiate radio frequency energy and if not installed in accordance with the operator's manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause interference in which case the user will be required to correct the interference at his own expense.

WARNING: Changes or modifications made to this device which are not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

DOC NOTICE

This digital apparatus does not exceed the Class A limits for radio noise emissions from digital apparatus set out in the Radio Interference Regulations of the Canadian Department of Communications.

Le présent appareil numérique n'émet pas de bruits radioélectriques dépassant les limites applicables aux appareils numériques de la class A prescrites dans le Règlement sur le brouillage radioélectrique édicté par le ministère des Communications du Canada.

VCCI NOTICE

This equipment is in the 1st Class Category (information equipment to be used in commercial and/or industrial areas) and conforms to the standards set by the Voluntary Control Council for Interference by Information Technology Equipment (VCCI) aimed at preventing radio interference in commercial and/or industrial areas.

Consequently, when used in a residential area or in an adjacent area thereto, radio interference may be caused to radios and TV receivers, etc.

Read the instructions for correct handling.

この装置は、第一種情報装置（商工業地域において使用されるべき情報装置）で商工業地域での電波障害防止を目的とした情報処理装置等電波障害自主規制協議会（VCCI）基準に適合しております。
従って、住宅地域またはその隣接した地域で使用すると、ラジオ、テレビジョン受信機等に受信障害を与えることがあります。
取扱説明書に従って正しい取り扱いをして下さい。

CABLETRON SYSTEMS, INC. PROGRAM LICENSE AGREEMENT

IMPORTANT: Before utilizing this product, carefully read this License Agreement.

This document is an agreement between you, the end user, and Cabletron Systems, Inc. ("Cabletron") that sets forth your rights and obligations with respect to the Cabletron software program (the "Program") contained in this package. The Program may be contained in firmware, chips or other media. BY UTILIZING THE ENCLOSED PRODUCT, YOU ARE AGREEING TO BECOME BOUND BY THE TERMS OF THIS AGREEMENT, WHICH INCLUDES THE LICENSE AND THE LIMITATION OF WARRANTY AND DISCLAIMER OF LIABILITY. IF YOU DO NOT AGREE TO THE TERMS OF THIS AGREEMENT, PROMPTLY RETURN THE UNUSED PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND.

CABLETRON SOFTWARE PROGRAM LICENSE

1. LICENSE. You have the right to use only the one (1) copy of the Program provided in this package subject to the terms and conditions of this License Agreement.

   You may not copy, reproduce or transmit any part of the Program except as permitted by the Copyright Act of the United States or as authorized in writing by Cabletron.

2. OTHER RESTRICTIONS. You may not reverse engineer, decompile, or disassemble the Program.

3. APPLICABLE LAW. This License Agreement shall be interpreted and governed under the laws and in the state and federal courts of New Hampshire. You accept the personal jurisdiction and venue of the New Hampshire courts.

EXCLUSION OF WARRANTY AND DISCLAIMER OF LIABILITY

1. EXCLUSION OF WARRANTY. Except as may be specifically provided by Cabletron in writing, Cabletron makes no warranty, expressed or implied, concerning the Program (including its documentation and media).

   CABLETRON DISCLAIMS ALL WARRANTIES, OTHER THAN THOSE SUPPLIED TO YOU BY CABLETRON IN WRITING, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WITH RESPECT TO THE PROGRAM, THE ACCOMPANYING WRITTEN MATERIALS, AND ANY ACCOMPANYING HARDWARE.

2. NO LIABILITY FOR CONSEQUENTIAL DAMAGES. IN NO EVENT SHALL CABLETRON OR ITS SUPPLIERS BE LIABLE FOR ANY DAMAGES WHATSOEVER (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF BUSINESS, PROFITS, BUSINESS INTERRUPTION, LOSS OF BUSINESS INFORMATION, SPECIAL, INCIDENTAL, CONSEQUENTIAL, OR RELIANCE DAMAGES, OR OTHER LOSS) ARISING OUT OF THE USE OR INABILITY TO USE THIS CABLETRON PRODUCT, EVEN IF CABLETRON HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. BECAUSE SOME STATES DO NOT ALLOW THE EXCLUSION OR LIMITATION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES, OR ON THE DURATION OR LIMITATION OF IMPLIED WARRANTIES, IN SOME INSTANCES THE ABOVE LIMITATIONS AND EXCLUSIONS MAY NOT APPLY TO YOU.

UNITED STATES GOVERNMENT RESTRICTED RIGHTS

The enclosed product (a) was developed solely at private expense; (b) contains “restricted computer software” submitted with restricted rights in accordance with Section 52227-19 (a) through (d) of the Commercial Computer Software - Restricted Rights Clause and its successors, and (c) in all respects is proprietary data belonging to Cabletron and/or its suppliers.

For Department of Defense units, the product is licensed with “Restricted Rights” as defined in the DoD Supplement to the Federal Acquisition Regulations, Section 52.227-7013 (c) (1) (ii) and its successors, and use, duplication, disclosure by the Government is subject to restrictions as set forth in subparagraph (c) (1) (ii) of the Rights in Technical Data and Computer Software clause at 252.227-7013. Cabletron Systems, Inc., 35 Industrial Way, Rochester, New Hampshire 03867-0505.

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 Document Conventions 1-2
1.2 Related Documentation 1-3
1.3 Getting Help 1-4
1.4 SNMP Primitives 1-4
1.5 MIB Primitive Types 1-5
1.6 User Functions 1-6
1.7 Navigating Through the MIB Tree Structure 1-7
1.8 TFTP 1-10
1.8.1 Software Distribution 1-10
1.8.2 Retrieving All Parameters 1-11
1.8.3 Setting All Parameters 1-11
1.9 Additional Interfaces 1-11

CHAPTER 2 TCP/IP MIB-II

2.1 System Group 2-1
2.2 Interfaces Group 2-3
2.3 Address Translation Group 2-8
2.4 IP Group 2-9
2.4.1 IP Address Table 2-11
2.4.2 IP Routing Table 2-13
2.4.3 IP Address Translation Table 2-16
2.4.4 IP Additional Objects 2-17
2.5 ICMP Group 2-17
2.6 TCP Group 2-20
2.7 UDP Group 2-21
2.7.1 UDP Listener Table 2-21
2.8 EGP Group 2-22
2.9 OIM Subtree 2-22
2.10 Transmission Group 2-22
2.11 SNMP Group 2-23

CHAPTER 3 ETHERNET MIB

3.1 Generic Ethernet-Like Group 3-1
3.2 Ethernet-Like Statistics Group 3-3
3.3 Ethernet-Like Collision Statistics Group 3-5
3.4 Ethernet-Like Tests Group 3-5
3.5 Ethernet-Like Errors Group 3-5
3.6 Ethernet-Like Chipsets Group 3-6

CHAPTER 4 TOKEN RING MIB

4.1 Token Ring Group....4-1
4.1.1 Token Ring Interface Table....4-1
4.1.2 Token Ring Statistics Table....4-4
4.1.3 Token Ring Timer Table....4-7
4.1.4 Token Ring Tests....4-7
4.1.5 Token Ring Chipsets....4-7

CHAPTER 5 FDDI MIB

5.1 SMT Group....5-1
5.1.1 SMT Table....5-2
5.2 MAC Group....5-6
5.2.1 MAC Table....5-6
5.3 MAC Counter Group....5-12
5.4 PATH Group....5-12
5.4.1 PATH Table....5-12
5.4.2 PATH Config Table....5-14
5.5 PORT Group....5-15
5.5.1 PORT Table....5-15

CHAPTER 6 BRIDGE MIB

6.1 Base Group....6-1
6.2 Spanning Tree Group....6-3
6.3 Source Route Group....6-8
6.4 Transparent Group....6-12
6.5 Static Database Group....6-15

CHAPTER 7 REPEATER MIB

7.1 Basic Package....7-1
7.2 Group Info....7-3
7.3 Port Info....7-5
7.4 Monitor Package....7-7
7.5 Monitor rptr Info....7-7
7.6 Monitor Group Info....7-7
7.7 Monitor Port Info....7-8
7.8 Address Tracking Package....7-13
7.9 Address Tracking rptr Info....7-13
7.10 Address Tracking Group Info....7-13
7.11 Address Tracking Port Info....7-13

CHAPTER 8 PPP MIB

8.1 PPP Link Control Table 8-1
8.2 PPP IP Table 8-9
8.3 PPP Bridge Support 8-11

CHAPTER 9 APPLETALK MIB

9.1 LLAP Group 9-1
9.2 AARP Group 9-3
9.3 ATPort Group 9-4
9.4 DDP Group 9-6
9.5 RTMP Group 9-8
9.6 ZIP Group 9-10
9.7 NBP Group 9-11
9.8 ATEcho Group 9-12

CHAPTER 10 CABLETRON MIB

10.1 System Group 10-1
10.2 Reserved Group 10-2
10.3 ATX Group 10-2
10.3.1 Hardware Configuration Group 10-3
10.3.2 Software Configuration Group 10-13
10.3.3 Unit Administration Group 10-15
10.3.3.1 Configuration Status Group 10-15
10.3.3.2 LMA Configuration Group 10-19
10.3.3.3 PPE Configuration Group 10-20
10.3.3.4 Spanning Tree Group 10-26
10.3.3.5 Mesh Group 10-27
10.3.4 Software Distribution Group 10-28
10.3.5 Addresses Configuration Group 10-30
10.3.6 SNMP/SMT Support Group 10-40
10.3.7 Cabletron Interfaces Group 10-42
10.3.8 Cabletron FDDI Interfaces Group 10-51
10.3.9 Cabletron UART Interface Group 10-56
10.3.10 Filter Group 10-57
10.3.11 Change Reboot Characteristics Command Group 10-64
10.3.12 Debug Command Group 10-66
10.3.13 Loopback Command Group 10-68
10.3.14 Cabletron WAN Interface Group 10-72
10.3.15 Cabletron Repeater Interface Group 10-74
10.3.16 Cabletron Repeater Interface Group 10-74
10.3.17 Cabletron Protocol Group 10-76

10.3.18 Cabletron IPX Group 10-81
10.3.18.1 SIPXIF Table 10-81
10.3.18.2 Route Table 10-83
10.3.18.3 SAP Table 10-84
10.3.19 Router Discovery Group 10-85
10.3.20 IP Multicasting Group 10-87
10.3.20.1 IP Multicast Neighbor Table 10-88
10.3.20.2 IP Multicast Stat Table 10-89
10.3.21 IP Circuit Table 10-89
10.3.22 sipNetToMedia Table 10-91
10.3.23 Trunking Group 10-91
10.3.24 IP Multicast Route 10-94
10.3.24.1 IPM Route Next Hop Table 10-96
10.3.24.2 IPM Route Interface Table 10-97
10.3.25 IGMP Interface 10-98
10.4 AppleTalk Group 10-100

CHAPTER 11 TRAPS

11.1 Generic Traps 11-1
11.2 Priority 0 Traps 11-2
11.3 Priority 1 Traps 11-3
11.4 Priority 2 Traps 11-8

APPENDIX A PACKET TRANSLATION PROCEDURE

CHAPTER 1
INTRODUCTION

This manual is for system administrators responsible for configuring, monitoring, and maintaining the ATX. Much of the configuration of the ATX needs to be done using an SNMP-based network management station. This manual contains the SNMP MIB variables you may need to configure, monitor, and manage your unit. You should use this manual with the *ATX User Guide* and with the documentation provided with your NMS.

The contents of each chapter are described below.

- Chapter 1, **Introduction**, provides an overview of the SNMP primitives, describes the functions the MIB variables can be used to perform, and describes how to use TFTP to download the ATX system software.
- Chapter 2, **TCP/IP MIB-II**, describes the standard TCP/IP MIB variables.
- Chapter 3, **Ethernet MIB**, describes the standard Ethernet MIB variables.
- Chapter 4, **Token Ring MIB**, describes the Token Ring MIB variables.
- Chapter 5, **FDDI MIB**, describes the FDDI MIB variables.
- Chapter 6, **Bridge MIB**, describes the Bridge MIB variables.
- Chapter 7, **Repeater MIB**, describes the Repeater MIB variables.
- Chapter 8, **PPP MIB**, describes the PPP link control and IP table MIB variables.
- Chapter 9, **AppleTalk MIB**, describes the AppleTalk MIB variables.
- Chapter 10, **Cabletron MIB**, describes the Cabletron enterprise MIB variables.
- Chapter 11, **Traps**, describes generic and enterprise-specific traps.
- Appendix A, **Packet Translation Procedure**, describes the canonical format the ATX uses for translating packets.

1.1 DOCUMENT CONVENTIONS

The following conventions are used in presenting information in this manual:

LCM commands, prompts, and information displayed by the computer appear in Courier typeface.

Note symbol. Calls the reader's attention to any item of information that may be of special importance.

Tip symbol. Conveys helpful hints concerning procedures or actions.

Caution symbol. Contains information essential to avoid damage to the equipment.

Warning symbol. Warns against an action that could result in equipment damage, personal injury or death.

1.2 RELATED DOCUMENTATION

You may need to refer to the following documentation:

- *ATX User Guide* – contains installation, configuration, and management instructions for the ATX. It also describes how to use the Local Console Manager (LCM), which is a non-intelligent terminal interface to the unit.

If you need internetworking reference material, you may find the following books helpful:

- *Interconnections, Bridges and Routers,* Radia Perlman, Addison Wesley © 1992.
- *Internetworking with TCP/IP: Principles, Protocols, and Architecture* (2nd edition), Volumes I and II, Douglas Comer, Prentice Hall © 1991.
- *The Simple Book, An Introduction to Management of TCP/IP-based internets* (2nd edition), Marshall T. Rose, Prentice Hall © 1994.

This manual describes the software interface between the NMS and the ATX. This is relevant for a unit running Version 1.0 software. The NMS communicates with the ATX software.

The Network Management, or UART, port is the interface to the Local Console Manager (LCM). LCM is a non-intelligent terminal interface that can be used to configure and monitor status for the unit.

1.3 GETTING HELP

If you need additional support related to this device, or if you have any questions, comments, or suggestions concerning this manual, contact Cabletron Systems Technical Support:

By phone		(603) 332-9400 Monday – Friday; 8 A.M. – 8 P.M. Eastern Time
By CompuServe		GO CTRON from any ! prompt
By Internet mail		support@ctron.com
By FTP		ctron.com (134.141.197.25)
	Login	*anonymous*
	Password	*your email address*

1.4 SNMP PRIMITIVES

The major software interface between the NMS and ATX consists of one simple mechanism – the exchange of SNMP (Simple Network Management Protocol, RFC 1157) datagrams over any available physical media. The following restrictions apply:

- All datagrams must obey SNMP format.
- All datagrams must be sent via UDP and IP. Thus, all datagrams will have UDP and IP headers.
- Datagrams may be sent over any of the following physical media:
    - Ethernet/802.3 LAN - the datagram must have an Ethernet MAC header, with an Ethernet frame type of IP; or, the datagram must be in 802.3 format with IP-encapsulation as defined by RFC 1042.
    - UART (out-of-band management port) - the datagram must have a PPP header, which indicates that the datagram contains an IP packet. (The unit automatically detects the presence of a PPP connection versus being connected to a non-intelligent terminal.)
- The NMS must rely on IP, rather than MAC addressing for all datagrams sent to the unit.
- All datagrams from the unit are addressed to either an NMS or the broadcast IP address.

Within the context of this document, the terms “datagram”, “packet”, and “PDU” are synonymous.

1.5 MIB PRIMITIVE TYPES

The MIB definitions in this document may reference the primitive types that are described in the Structure and Identification of Management Information for TCP/IP-based Internets, RFC 1155. RFC 1155 is based on the Specification of Abstract Syntax Notation One, ASN.1. The primitive types are described in Table 1-1.

Table 1-1 Primitive Descriptions

Primitive	Size	Description
Boolean	1 byte	Enumerated Integer with possible true (1) or false (2) values; note that the ASN.1 BOOLEAN primitive type is not used
BridgeID	8 bytes	Priority and MAC address used to identify a spanning tree bridge
Counter	4 bytes max	Unsigned value
DisplayString	n X1 byte	Array of printable ascii characters
Gauge	4 bytes	Non-negative integer
Integer	4 bytes max	Signed value
IpAddress	4 bytes	Internet address
MacAddress	6 bytes	Ethernet address
OctetString	n X1 byte	Array of bytes
PhysAddress	n X1 byte	Array of bytes, using the same as a MAC Address
PortID	2 bytes	Priority and port number used to identify a spanning tree port
TimeTicks	4 bytes	Max time counter with a granularity of 1/100th of a second (also known as centiseconds)

1.6 USER FUNCTIONS

The SNMP primitives may be used to accomplish the following functions:

- Obtain the unit's current value of certain parameters - the NMS uses the GetRequest or GetNextRequest PDU, and the unit responds with a GetResponse PDU. If the NMS issues a GetRequest for an unsupported parameter, the unit sends a GetResponse with a noSuchName ErrorStatus[1]. If the NMS issues a GetNextRequest for an unsupported parameter, the unit skips to the next object.
- Change the unit's value of certain parameters - the NMS uses the SetRequest PDU, and the unit responds with a GetResponse PDU. The unit will change both its current value and its local default to be used when the unit reboots, unless noted otherwise.
- Obtain the current value of certain parameters and simultaneously change the value of other parameters - the NMS uses the SetRequest PDU, and the init responds with a GetResponse PDU. For the parameters which are being obtained rather than changed, the NMS must use the ASN.1 NULL value with the SetRequest PDU.
- Provide notification of significant events - the unit uses the Trap PDU.

The unit implements two non-standard features with respect to the SNMP SetRequest:

- The variable bindings within a SetRequest are sometimes processed sequentially rather than simultaneously. For example, if a SetRequest contains two parameters with an incorrect value specified for the second parameter, the unit returns a badValue error to the NMS; however, the unit may have updated its value for the first parameter.
- The values within the variable bindings of the returned GetResponse may reflect meaningful information, rather than being an exact copy of the values from the SetRequest. For example, if a SetRequest contains two variable bindings, the first specifying that memory should be examined and the second specifying the contents of the memory, then the unit's GetResponse will update the value of the second variable binding to contain the actual contents of the memory.

---

1. If implementing the parameter is required, it might seem more reasonable to return a GetResponse with no error and the ASN.1 NULL value as the parameter's value; however, leading authorities such as Marshall T. Rose (author of *The Simple Book*) suggest that noSuchName be returned, because many existing SNMP management stations do not handle NULL values correctly.

Many parameters are described by existing Management Information Base (MIB) documents. The meanings of those parameters are not always implemented exactly by the unit; instead, the parameters are treated as best as can be correlated to the unit's parameters. The unit's parameters that cannot be correlated are part of a Cabletron proprietary MIB, called: [2]

```
sigma OBJECT IDENTIFIER = {enterprises 97}
```

The parameters and traps which are supported are described in the chapters that follow.

1.7 NAVIGATING THROUGH THE MIB TREE STRUCTURE

The MIB structure is a hierarchical tree structure. Each MIB variable has a numeric value that indicates its place in the hierarchy. The structure was originally created, and is still maintained by the International Organization for Standardization (ISO) and the International Telecommunications Union (ITU), two international standards organizations. You can get and set MIB variables by navigating down the tree to a specific MIB, a group or table within that MIB, and then to the individual variable. Figure 1-1 shows the path down the MIB tree structure. Under the "mib-2" and the Cabletron enterprise branch, are all the relevant MIBs that the unit supports.

2. Usually, it is better to support a standard parameter in a non-standard fashion, than not support the standard parameter at all. Deciding what constitutes an acceptable deviation is a gray area.

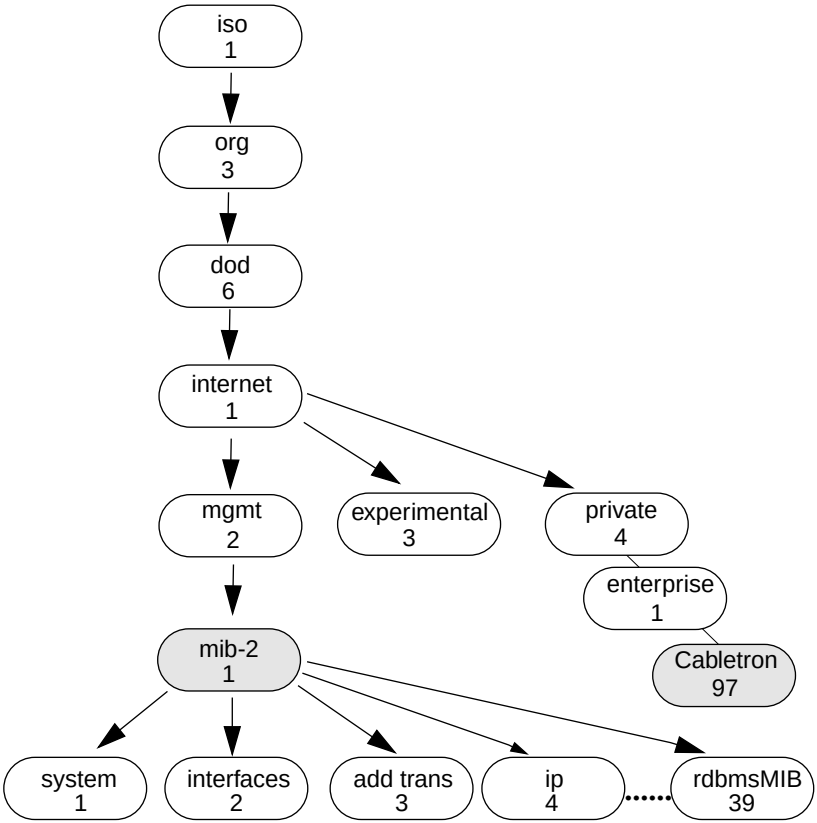


Figure 1-1 MIB Hierarchical Structure

Table 1-2 provides the branch structure that is under MIB-II. To reach any of the MIB-II objects you would start with the prefix 1.3.6.1.2.1. For example, to reach an object in the system group, you would start with 1.3.6.1.2.1.1. To find the amount of time the ATX had been running, you would want to get the sysUpTime variable, which is the third object in the system group. So the get command would look like:

```
get 1.3.6.1.2.1.1.3.0
```

The zero at the end (.0), indicates that this is a single instance, and that only one value can be returned. Whenever you are looking for a variable with only one value, you must include the .0 at the end. Some variables may have multiple values, such as an IP address and an associated port number.

Table 1-2 Branches Under MIB-II

MIB-II Group	Number (1.3.6.1.2.1.)
System	1
Interfaces	2
Address Translation	3
Internet Protocol (IP)	4
Internet Control Message (ICMP)	5
Transmission Control Protocol (TCP)	6
User Datagram Protocol (UDP)	7
Exterior Gateway Protocol (EGP)	8
CMIP over TCP (CMOT)	9
Transmission	10
SNMP	11
GenericIF	12
AppleTalk	13
Open Shortest Path First (OSPF)	14
Border Gateway Protocol (BGP)	15
Remote Network Monitoring (RMON)	16
Bridge	17

The Cabletron MIB is under the private enterprise MIB branch. To identify a variable in the Cabletron MIB, you would start with the private enterprise prefix of 1.3.6.1.4.1, and add the specific Cabletron Massachusetts ID of 97. The result, 1.3.6.1.4.1.97. , is the complete prefix for an Cabletron MIB variable. You would then add the specific object ID to complete the MIB variable.

For example, to find the number of filters currently defined in the ATX, you would want to get the filterCurrentCount variable in the Cabletron MIB filter group and add it to the prefix 1.3.6.1.4.1.97. After the prefix, add the ATX group ID {2}, the filter group ID {9}, and the filterCurrentCount ID of {2}. As stated above, the zero {0} indicates that this variable is a single instance and only one variable can be returned.

The get command would look like:

```
get 1.3.6.1.4.1.97.2.9.2.0
```

At the beginning of each chapter in this Reference Guide, the prefix for each MIB group is provided. To calculate the specific MIB variable, you add the specific object ID to the prefix for that MIB group.

1.8 TFTP

TFTP (Trivial File Transfer Protocol, RFC 1350) is used for:

- Distribution of new software.

TFTP has no inherent security provision; however, all files have special data encryption, and the ATX will reject files that have not been encrypted. In addition, SNMP primitives may be used to prevent the ATX from accepting unauthorized TFTP requests, even if the files have the special data encryption. Refer to the description of the swdis branch of Cabletron's private MIB for TFTP security details.

1.8.1 Software Distribution

TFTP is used for the distribution of new software. The new software will be automatically invoked when a ATX reboots.

New software is released in two files:

- dnld_hdr
- dnld_software

To distribute the new software to a ATX, the TFTP procedure is as follows:

1. Start TFTP on the NMS, or on any other device which can provide TFTP services. (Typically, TFTP must be started from the same directory that contains the files to be transferred.) TFTP must be told the IP address of the remote host (the IP address of the ATX), and the file transfer mode (which must be "binary").
2. Use TFTP to retrieve all of the unit's parameters, as descibed in a subsection below. This step is not required, but it is recommended if you need to back-pedal to the software that is currently being executed by the unit.
3. Tell TFTP to transfer ("put") the first file, (dnld_hdr).
4. Wait at least three minutes, or until the unit sends the SNMP Trap described in the swdis branch of Cabletron's private MIB.

5. Tell TFTP to transfer the second file, (dnld_software). This transfer should take approximately six minutes.

If the above TFTP sequence is abnormally terminated, there is no cause for alarm, since the ATX maintains a back-up set of software, and the ATX will not use the incomplete new software.

1.8.2 Retrieving All Parameters

TRTP is used for retrieval of the parameters of an ATX, as follows:

1. Start TFTP. TFTP must be told the IP address of the ATX, and the file transfer mode (binary).
2. Tell TFTP to retrieve the unit's Configuration file.
3. After about ten seconds, the TFTP operation will be complete.

1.8.3 Setting All Parameters

TFTP is used for bulk setting of all of the parameters of an ATX, as follows:

1. Start TFTP. TFTP must be told the IP address of the unit, and the file transfer mode (binary).
2. Tell TFTP to retrieve the unit's Configuration file.
3. After about ten seconds, the TFTP operation will be complete.

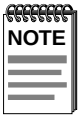
1.9 ADDITIONAL INTERFACES

In addition to SNMP and TFTP, the ATX employs the following protocols, as part of its software interface with an NMS:

- UDP - User Datagram Protocol, RFC 768.
- IP - Internet Protocol, RFC 791.
- ARP - Ethernet Address Resolution Protocol, RFC 826.
- RARP - Reverse Address Resolution Protocol, RFC 903. RARP is only used when no IP addresses have been assigned to the ATX.
- BOOTP - Bootshop Protocol, RFC 951 (will be used as an alternative to RARP, so that the ATX can learn IP Addresses across networks connected by IP routers that perform the BOOTP function).

CHAPTER 2
TCP/IP MIB-II

The ATX supports the TCP/IP MIB-II, as defined by *Management Information Base for Network Management of TCP/IP-Based Internets MIB-II*, RFC 1213 (K. McCloghrie, editor), dated March 1991. The MIB is divided into groups of parameters. The individual groups are described in the subsections below. You may want to refer to the actual TCP/IP MIB, since this document paraphrases the standard MIB in order to provide ATX-related descriptions.



The access types of some of the TCP/IP MIB parameters have been extended from *Read-Only* to *Read-Write*. However, unless otherwise indicated, the access type of all parameters is as indicated in the standard MIB.

The MIB tree prefix for reaching the TCP/IP MIB-II is:

```
1.3.6.1.2.1.
```

2.1 SYSTEM GROUP

system {mib-2 1}

The TCP/IP System Group parameters are described below.

sysDescr {system 1}
DisplayString Read-Only

A textual description of the unit, including the full name and version identification of the software and hardware (refer to swDesc and hwManufData). The text obeys the format “ZAHMHTHOH1H2H3H4”, where,

A — a 0-32 character textual description of the current set of software files. The current set of files is used whenever the unit boots or an IOM is reloaded.

HM — the name of the board occupying the Packet Processing Engine slot, followed by 2 blanks and a 24 character manufacturing code (which contains a part number and serial number).

HT	the name TURBO followed by 24 characters which are all blanks. If the board is removable, then the 24 characters contain a part number and serial number. If the board is revocable and has been removed, then the name is NONE followed by 25 blanks.
Hn	the name of each IOM followed by a 24 character manufacturing code (which contains a part number and serial number). Each IOM name is padded with blanks, if necessary, so that it is 5 characters. The IOM names are the following.
CEIOM	Ethernet or 802.3 (i.e., 10 megabit CSMA/CD) module with a built-in concentrator
EIOM	Ethernet or 802.3 module
FIOM	FDDI module
HIOM	HSSI module
IFIOM	Intelligent FDDI module
NONE	no module (the manufacturing code is all blanks)
TIOM	Token Ring (i.e., 802.5) module
QEIOM	Quad Ethernet module, i.e., an Ethernet module with 4 ports
QTIOM	Quad Token Ring module, i.e., a Token Ring module with 4 ports

sysObjectID {system 2}
Object Identifier Read-Only

The identifier of the variable used to identify the type of entity. Cabletron's MIB sysID parameter will have a value that indicates that it is a hub/bridge/router. So sysObjectID contains the object identifier of the Cabletron MIB sysID object, i.e.,
{1 3 6 1 4 1 97 1 1}.

sysUpTime {system 3}
TimeTicks Read-Only

The time, in centiseconds, since the unit was last booted.

sysContact {system 4}
DisplayString Read-Write

The name and address of the contact person for the unit. The first 24 characters of sysContact are copied as the UserData information for FDDI SMT frames.

sysName {system 5}
DisplayString Read-Write

An administratively-assigned name for this node. By convention, this is the node's fully qualified domain name. The LCM prompt consists of sysName followed by ">". (LCM is a non-intelligent terminal interface that can be used to configure and monitor status for the unit.)

sysLocation {system 6}
DisplayString Read-Write

The physical location of the unit.

sysServices {system 7}
Integer Read-Only

The sum of the services supported by the unit. Values include:

- (2) - the unit is a bridge only
- (4) - the unit is an IP router only
- (6) - the unit is both a bridge and an IP router

The unit is a bridge and IP router.

2.2 INTERFACES GROUP

interfaces {mib-2 2}

The TCP/IP Interfaces Group parameters are described below.

ifNumber {interfaces 1}
Integer Read-Only

The number of ports (whether alive or dead), including the UART, as indicated by the Configuration EEPROM. The actual ports (as indicated by the types of the boards currently present in the IOM slots) are not part of this MIB parameter; however, the unit tends to automatically adjust its Configuration EEPROM to match the boards that are currently present.

ifTable {interfaces 2}
Not-Accessible

A list of interface entries; one per port (ifNumber in total).

ifEntry {ifTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

ifIndex {ifEntry 1}
Integer Read-Only

The port number, beginning with 1 for the first port. This number always matches the instance of the ifEntry. For example, {ifIndex 3} contains the value “3”.

ifDescr {ifEntry 2}
DisplayString Read-Only

A textual description of the port. This contains the “Hn” of the port’s IOM, as described for the sysDescr parameter. [For the UART, this contains the “HM” portion of sysDescr.]

ifType {ifEntry 3}
Integer Read-Only

The type of the port, i.e., one of the following:

- (6) - ethernet-csmacd (for CEIOM, EIOM, and QEIOM ports)
- (9) - iso88025 tokenRing (for TIOM and QTIOM ports)
- (15) - fddi (for FIOM and IFIOM ports)
- (23) - ppp (for the UART and HIOM ports).

ifMtu {ifEntry 4}
Integer Read-Write

The size (in bytes) of the largest network datagram which may be sent or received on the port. This does not include the MAC header, LLC header, and FCS. For CSMA/CD ports, the Ethernet Frame Type is considered part of the MAC header, but there is no LLC header. For PPP ports, changing ifMtu automatically causes sifMaxPacketSize and pppLinkLocalMRU to change. Specifying a value of zero in a SetRequest indicates that the interface is to default to the largest MTU available for that media. The effective value is always returned in the GetResponse. This is useful primarily for 802.5 Token Ring, where the allowable MTU is different for 4 Mbps and 16 Mbps rings.[1]

[1.] The standard MIB definition is Read-Only.

ifSpeed {ifEntry 5}
Integer Read-Only

The port's estimated MAC-level bandwidth, in bits per second. The bandwidth will be in the range 1,200 - 100,000,000.

ifPhysAddress {ifEntry 6}
Physical Address Read-Write

The MAC address of the port. For the UART and WANs, this field should be an octet string containing a single octet with a value of hexadecimal FF.

ifAdminStatus {ifEntry 7}
Integer Read-Write

The desired state of the port, i.e., one of the following

- up (1) - the port is to be used for sending/receiving NMS packets; however, whether or not the port has its bridging and/or routing functions enabled is unknown (for the UART, the value of ifAdminStatus is always restored to "up" when the unit reboots).
- down (2) - the port is not to be used
- testing (3) - the port is to be put into local loopback (this value is not saved if the unit reboots).

ifOperStatus {ifEntry 8}
Integer Read-Only

The current status of the port, i.e., one of the following:

- up (1) - the port can send/receive NMS packets; however, whether or not the port has its bridging functions and/or routing functions enabled is unknown
- down (2) - the port is broken, or is intentionally physically disabled
- testing (3) - the port is in local loopback.

ifLastChange {ifEntry 9}
TimeTicks Read-Only

The time, in centiseconds relative to sysUpTime, since the port entered its last state (as defined by ifOperStatus).

ifInOctets {ifEntry 10}
Counter Read-Only

The total number of bytes received on the port, counting the MAC header and FCS, but not counting the bytes in packets that were rejected due to hardware errors. **All counters are 32 bit wide wrap-around counters which can only be reset by restarting the port or by rebooting the unit.**

ifInUcastPkts {ifEntry 11}
Counter Read-Only

The number of non-multicast packets received by the port, regardless of the packet's outcome (i.e., whether the packet was filtered or forwarded).

ifInNUcastPkts {ifEntry 12}
Counter Read-Only

The number of multicast packets received by the port, regardless of the packet's outcome (i.e., whether the packet was filtered or forwarded).

ifInDiscards {ifEntry 13}
Counter Read-Only

The number of packets received by the port, which were filtered (by the unit's forward/filter decision-making).

ifInErrors {ifEntry 14}
Counter Read-Only

The number of packets received by the port, which were discarded due to hardware reception errors.

ifInUnknownProtos {ifEntry 15}
Counter Read-Only

The number of packets received by the port, which were discarded because of an unknown or unsupported protocol.

ifOutOctets {ifEntry 16}
Counter Read-Only

The total number of bytes transmitted out the port, counting the MAC header and FCS.

ifOutUcastPkts {ifEntry 17}
Counter Read-Only

The number of non-multicast packets transmitted out the port, regardless of whether or not hardware transmission errors were encountered.

ifOutNUcastPkts {ifEntry 18}
Counter Read-Only

The number of multicast packets transmitted out the port, regardless of whether or not hardware transmission errors were encountered.

ifOutDiscards {ifEntry 19}
Counter Read-Only

The number of packets to be transmitted out the port, but were not transmitted due to non-error reasons. The definition of non-error reasons is implementation dependent. The unit defines non-error reasons as packet congestion. Packet congestion occurs when too many packets are to be queued for transmission, or when packets have been awaiting transmission for too long a time period.

ifOutErrors {ifEntry 20}
Counter Read-Only

The number of packets that were to be transmitted out the port, but incurred transmission hardware errors.

ifOutQLen {ifEntry 21}
Gauge Read-Write

The maximum length ever obtained by the port's outbound packet queue (in packets). The size of a port's outbound packet queue is periodically polled (rather than re-calculated whenever a packet is added), so ifOutQLen is an approximation of the maximum length.[2]

ifSpecific {ifEntry 22}
Object Identifier Read-Only

The object identifier of the MIB for the type of port, i.e., one of the following:

- {dot3} - for Ethernet ports (dot3 is defined later, as {transmission 7}).
- {ppp} - for the UART and WAN ports (ppp is defined later, as {experimental 18}).

---

2. The standard MIB definition if Read-Only.

- {fddi} - for FDDI ports (fddi is defined later, as {transmission 15}).
- {dot5} - for Token Ring ports (dot5 is defined later, as {transmission 9}).

2.3 ADDRESS TRANSLATION GROUP

at {mib-2 3}

The TCP/IP Address Translation Group parameters are minimally supported (i.e., for any parameter in this group, the unit returns a GetResponse with a noSuchName ErrorStatus). It is anticipated that the Address Translation Group will be deleted from TCP/IP MIB III, since there will be separate address translation tables for every type of network protocol (indeed, TCP/IP MIB II already defines the IP Address Translation Table).

The TCP/IP Address Translation Group parameters are described below.

atTable {at 1}
Not-Accessible

atEntry {atTable 1}
Not-Accessible

atIfIndex {atEntry 1}
Integer Read-Write

All GetResponse PDUs indicate a noSuchName ErrorStatus.

atPhysAddress {atEntry 2}
Physical Address Read-Write

All GetResponse PDUs indicate a noSuchName ErrorStatus.

atNetAddress {atEntry 3}
Network Address Read-Write

All GetResponse PDUs indicate a noSuchName ErrorStatus.

2.4 IP GROUP

ip {mib-2 4)

The TCP/IP IP Group parameters are described below.

ipForwarding {ip 1}
Integer Read-Write

Whether the unit is an IP router. This is always gateway (1), i.e., the unit is an IP router (or combination bridge/router). Attempts to write to this field with a different value will not change its value, and “badValue” GetResponse PDUs are returned for all such SetRequest PDUs.

ipDefaultTTL {ip 2}
Integer Read-Write

The value, in seconds, to insert into the Time-To-Live field of the IP header when this unit creates IP datagrams.

ipInReceives {ip 3}
Counter Read-Only

The total number of IP packets received from all ports (including the UART).

ipInHdrErrors {ip 4}
Counter Read-Only

The number of packets received that were discarded due to errors in the IP header.

ipInAddrErrors {ip 5}
Counter Read-Only

The number of packets received that were discarded due to an invalid (or nonroutable) destination IP address in the IP header.

ipForwDatagrams {ip 6}
Counter Read-Only

The number of packets received that were routed towards a final IP destination.

ipInUnknownProtos {ip 7}
Counter Read-Only

The number of packets received that were addressed to this unit’s IP, but were discarded because of an unknown or unsupported protocol.

ipInDiscards {ip 8}
Counter Read-Only

The number of packets that were received without error, but were not processed (due to insufficient resources, for example).

ipInDelivers {ip 9}
Counter Read-Only

The total number of input packets successfully delivered to the IP user-protocol layers.

ipOutRequests {ip 10}
Counter Read-Only

The total number of IP output packets generated by this unit. This count does not include any packets represented in ipForwDatagrams.

ipOutDiscards {ip 11}
Counter Read-Only

The total number of output packets which were discarded (due to lack of resources, for example). This counter includes packets which would be included in ipForwDatagrams if any such packets were discarded.

ipOutNoRoutes {ip 12}
Counter Read-Only

The number of packets which were discarded because no route could be found to transmit them to their destination. This counter includes any packets counted in ipForwDatagrams which meet this “no-route” criterion.

ipReasmTimeout {ip 13}
Integer Read-Only

The maximum time, in seconds, that received fragments are held while they are awaiting reassembly within this unit.

ipReasmReqds {ip 14}
Counter Read-Only

The number of IP fragments received which needed to be reassembled within this unit.

ipReasmOKs {ip 15}
Counter Read-Only

The number of IP datagrams which were successfully reassembled.

ipReasmFails {ip 16}
Counter Read-Only

The number of failures (for whatever reason timed-out, errors, etc.) detected by the IP reassembly algorithm. This is not necessarily a count of discarded IP fragments since some algorithms (notably RFC 815's) can lose track of the number of fragments by combining them as they are received.

ipFragOKs {ip 17}
Counter Read-Only

The number of IP datagrams that have been successfully fragmented within this unit.

ipFragFails {ip 18}
Counter Read-Only

The number of IP datagrams that have been discarded because they needed to be fragmented but could not be (e.g. because their "Don't Fragment" flag was set).

ipFragCreates {ip 19}
Counter Read-Only

The number of IP datagram fragments that have been generated by this unit.

2.4.1 IP Address Table

The TCP/IP IP Address Table contains the IP addressing information for each port. The parameters are described below.

ipAddrTable {ip 20}
Not-Accessible

A list of IP address entries; one per IP address. If a port has not yet learned its IP address, then an ipAddrEntry might not exist for the port (i.e., having an ipAddrEntry with an IP address of zero is not acceptable).

ipAddrEntry {ipAddrTable 1}
Not-Accessible

A set of objects for an ipAddrTable entry. The individual components are described below.

ipAdEntAddr {ipAddrEntry 1}
IP Address Read-Only

The IP address itself. Although this field is *Read-Only*, the unit's sifIpAddress parameter can be written to affect changes.

ipAdEntIfIndex {ipAddrEntry 2}
Integer Read-Only

The port number which has the indicated IP address.

ipAdEntNetMask {ipAddrEntry 3}
IP Address Read-Write[3]

The network/subnet mask associated with the IP address. For SetRequests, a mask of zero may be specified, in which case, the unit will create the correct network mask for a non-subnetted network. For GetRequests, the unit will always convert a value of zero into the correct network mask.

ipAdEntBcastAddr {ipAddrEntry 4}
Integer Read-Only

The value for the least significant bit for broadcasts, i.e., the constant 1.

ipAdEntReasmMaxSize {ipAddrEntry 5}
Integer Read-Only

The largest IP datagram which can be reassembled, i.e., the constant 4470.

3. The standard MIB definition is *Read-Only.*

2.4.2 IP Routing Table

The TCP/IP IP routing table contains the routing information for each route currently known by the unit. When adding a row, the entire row must be specified, except for the following defaults:

- ipRouteDest is obtained from the row's instance identifier
- ipRouteMetric1 defaults to 0
- ipRouteMetric2 defaults to ppeStaticPreference
- ipRouteMetric3 through ipRouteMetric5 default to -1
- ipRouteNextHop is obtained from the row's instance identifier
- ipRouteType defaults to "direct"
- ipRouteProto defaults to "netmgmt"
- ipRouteMask defaults to the standard network class mask based on the row's instance identifier (except 255.255.255.255 is used if the row's instance identifier contains a non-zero host portion)

Parameters include the following variables.

ipRouteTable {ip 21}
Not-Accessible

A list of routing entries; one per route. An entry is keyed by an IP address (ipRouteDest, defined below). If there are multiple entries for the same IP address, then only the entry being used by the IP forwarding process is available for SNMP access.

ipRouteEntry {ipRouteTable 1}
Not-Accessible

A set of objects for an ipRouteTable entry. The individual components are described below.

ipRouteDest {ipRouteEntry 1}
IP Address Read-Write

The destination IP address of this route.

ipRouteIfIndex {ipRouteEntry 2}
Integer Read-Write

The port number of the next hop.

ipRouteMetric1 {ipRouteEntry 3}
Integer Read-Write

The primary routing metric for this route. This is the zero-based hop count to ipRouteDest. That is, a hop count of zero indicates that ipRouteDest is on a directly connected network.

ipRouteMetric2 {ipRouteEntry 4}
Integer Read-Write

An alternative routing metric for this route. This must be greater than zero.

ipRouteMetric3 {ipRouteEntry 5}
Integer Read-Write

An alternative routing metric for this route.

ipRouteMetric4 {ipRouteEntry 6}
Integer Read-Write

An alternative routing metric for this route.

ipRouteNextHop {ipRouteEntry 7}
IP Address Read-Write

The IP address of the route's next hop.

ipRouteType {ipRouteEntry 8}
Integer Read-Write

The type of the route, one of the following:

- other (1) - none of the below
- invalid (2) - the entry should be considered to be non-existent
- direct (3) - the entry is a route to a directly connected network
- indirect (4) - the entry is a route to a remote network

ipRouteProto {ipRouteEntry 9}
Integer Read-Only

The manner in which the route was learned. Values include:

- other (1) - none of the following
- local (2) - non-protocol information, manually
- netmgmt (3) - configured entries set via a network management protocol
- icmp (4) - obtained via ICMP e.g., Redirect

The remaining values are all gateway routing protocols:

- egp (5)
- ggp (6)
- hello (7)
- rip (8)
- is-is (9)
- es-is (10)
- ciscoIgrp (11)
- bbnSpfIgp (12)
- ospf (13)
- bgp (14)

ipRouteAge {ipRouteEntry 10}
Integer Read-Write

The number of seconds since the route was last updated or deemed to be correct.

ipRouteMask {ipRouteEntry 11}
IP Address Read-Write

The mask to be ANDed with the destination IP address, before being compared to the value in the ipRouteDest field.

ipRouteMetric5 {ipRouteEntry 12}
Integer Read-Write

An alternative routing metric for this route.

ipRouteInfo {ipRouteEntry 13}
Object Identifier Read-Write

A reference to additional MIB definitions, specific to the routing protocol which is responsible for this route. This information is not present in the unit, so ipRouteInfo should be set to the object identifier {0 0}.

2.4.3 IP Address Translation Table

The TCP/IP IP address translation table contains mappings of Internet addresses to MAC addresses, except for the Internet addresses of the unit itself. The parameters are Read/Write, according to TCP/IP MIB-II; however, the unit has the restriction that a dynamic entry should not be modified, except to be made "invalid" (any other modifications will cause a static entry to be created). When adding a row, the following defaults are used:

- ipNetToMediaIfIndex is obtained from the row's instance identifier
- ipNetToMediaPhysAddress defaults to 0
- ipNetToMediaNetAddress is obtained from the row's instance identifier
- ipNetToMediaType defaults to "static"

Parameters include the following:

ipNetToMediaTable {ip 22}
Not-Accessible

A list of IP address translation entries; one per port number and IP address.

ipNetToMediaEntry {ipNetToMediaTable 1}
Not-Accessible

A set of objects for an ipNetToMediaTable entry. The individual components are described below.

ipNetToMediaIfIndex {ipNetToMediaEntry 1}
Integer Read-Write

The port number for which this entry is effective.

ipNetToMediaPhysAddress {ipNetToMediaEntry 2}
Physical Address Read-Write

The MAC address (for PPP ports, this should be a single byte of hexadecimal FF; for FDDI and Token Ring ports, this should be little endian, although there are some factions of the SNMP community which believe this should be big endian; for Ethernet ports, this should be little endian).

ipNetToMediaNetAddress {ipNetToMediaEntry 3}
IP Address Read-Write

The IP address which corresponds to the ipNetToMediaPhysAddress.

ipNetToMediaType {ipNetToMediaEntry 4}
Integer Read-Write

The type of address mapping, one of the following

- other (1) - none of the below
- invalid (2) - the entry should be considered to be non-existent
- dynamic (3) - the mapping was dynamically learned
- static (4) - the mapping was statically configured

2.4.4 IP Additional Objects

The TCP/IP additional IP objects are described below.

ipRoutingDiscards {ip 23}
Counter Read-Only

The number of valid routing entries that were discarded.

2.5 ICMP GROUP

icmp {mib-2 5}

The TCP/IP ICMP Group parameters are described below.

icmpInMsgs {icmp 1}
Counter Read-Only

The total number of ICMP messages which were received by this unit. This includes all messages represented by icmpInErrors.

icmpInErrors {icmp 2}
Counter Read-Only

The number of ICMP messages which were received with errors (bad checksums, bad length, etc.).

icmpInDestUnreachs {icmp 3}
Counter Read-Only

The number of ICMP “Destination Unreachable” messages received.

icmpInTimeExcds {icmp 4}
Counter Read-Only

The number of ICMP “Time Exceeded” messages received.

icmpInParmProbs {icmp 5}
Counter Read-Only

The number of ICMP “Parameter Problem” messages received.

icmpInSrcQuenchs {icmp 6}
Counter Read-Only

The number of ICMP “Source Quench” messages received.

icmpInRedirects {icmp 7}
Counter Read-Only

The number of ICMP “Redirect” messages received.

icmpInEchos {icmp 8}
Counter Read-Only

The number of ICMP “Echo (request)” messages received.

icmpInEchoReps {icmp 9}
Counter Read-Only

The number of ICMP “Echo Reply” messages received.

icmpInTimestamps {icmp 10}
Counter Read-Only

The number of ICMP “Timestamp (request)” messages received.

icmpInTimestampsReps {icmp 11}
Counter Read-Only

The number of ICMP “Timestamp Reply” messages received.

icmpInAddrMasks {icmp 12}
Counter Read-Only

The number of ICMP “Address Mask Request” messages received.

icmpInAddrMaskReps {icmp 13}
Counter Read-Only

The number of ICMP “Address Mask Reply” messages received.

icmpOutMsgs {icmp 14}
Counter Read-Only

The total number of ICMP messages which were sent by this unit. This includes all messages counted by icmpOutErrors.

icmpOutErrors {icmp 15}
Counter Read-Only

The number of ICMP messages which this unit did not send due to problems discovered entirely within the ICMP subsystem (such as lack of buffers).

icmpOutDestUnreachs {icmp 16}
Counter Read-Only

The number of ICMP “Destination Unreachable” messages sent.

icmpOutTImeExcds {icmp 17}
Counter Read-Only

The number of ICMP “Time Exceeded” messages sent.

icmpOutParmProbs {icmp 18}
Counter Read-Only

The number of ICMP “Parameter Problem” messages sent.

icmpOutSrcQuenchs {icmp 19}
Counter Read-Only

The number of ICMP “Source Quench” messages sent.

icmpOutRedirects {icmp 20}
Counter Read-Only

The number of ICMP “Redirect” messages sent.

icmpOutEchos {icmp 21}
Counter Read-Only

The number of ICMP “Echo (request)” messages sent.

icmpOutEchoReps {icmp 22}
Counter Read-Only

The number of ICMP “Echo Reply” messages sent.

icmpOutTimestamps {icmp 23}
Counter Read-Only

The number of ICMP “Timestamp (request)” messages sent.

icmpOutTimestampReps {icmp 24}
Counter Read-Only

The number of ICMP “Timestamp Reply” messages sent.

icmpOutAddrMasks {icmp 25}
Counter Read-Only

The number of ICMP “Address Mask Request” messages sent.

icmpOutAddrMaskReps {icmp 26}
Counter Read-Only

The number of ICMP “Address Mask Reply” messages sent.

2.6 TCP GROUP

tcp {mib-2 6}

The TCP Group is not supported since the unit does not support TCP.

2.7 UDP GROUP

udp {mib-2 7}

The TCP/IP UDP Group parameters are described below.

udpInDatagrams {udp 1}
Counter Read-Only

The total number of UDP datagrams delivered to UDP users.

udpNoPorts {udp 2}
Counter Read-Only

The total number of received UDP datagrams for which there was no application at the destination port.

udpInErrors {udp 3}
Counter Read-Only

The number of received UDP datagrams that could not be delivered for reasons other than the lack of an application at the destination port.

udpOutDatagrams {udp 4}
Counter Read-Only

The total number of UDP datagrams sent from this unit.

2.7.1 UDP Listener Table

The TCP/IP UDP Listener Table parameters are described below.

udpTable {udp 5}
Not-Accessible

A table containing UDP listener information. The table has entries for the following UDP port numbers port 69 (default TFTP port number), port 161 (SNMP port number), and port 520 (RIP port number). In the future, an additional table entry may exist for a TFTP session that has been established with other than the TFTP default UDP port number.

udpEntry {udpTable 1}
Not-Accessible

The entry information of the IP address and UDP port combination.

udpLocalAddress {udpEntry 1}
IP Address Read-Only

The all zeroes IP address (0.0.0.0), which indicates that the UDP listener is willing to accept UDP datagrams for any IP address associated with the unit.

udpLocalPort {udpEntry 2}
Integer Read-Only

The UDP port number, i.e., one of 69, 161, and 520. Note that reception of SNMP Traps (UDP port number 162) is not supported by the unit.

2.8 EGP GROUP

egp {mib-2 8}

The EGP Group is not supported since the unit does not support EGP.

2.9 OIM SUBTREE

cmot {mib-2 9}

The OSI Subtree, as defined in *OSI internet management Management Information Base*, RFC 1214 (L. LaBarre, editor), dated April 1991, is not supported.

2.10 TRANSMISSION GROUP

transmission {mib-2 10}

The supported TCP/IP Transmission Group parameters are described in later sections of this document. The transmission group supported is:

dot3 {transmission 7}

dot5 {transmission 9}

fddi {transmission 15}

ppp {transmission 23}

2.11 SNMP GROUP

snmp {mib-2 11}

The TCP/IP SNMP Group parameters are described below.

snmpInPkts {snmp 1}
Counter Read-Only

The number of SNMP PDUs received by the unit.

snmpOutPkts {snmp 2}
Counter Read-Only

The number of SNMP PDUs created by the unit.

snmpInBadVersions {snmp 3}
Counter Read-Only

The number of SNMP PDUs received by the unit which had an unsupported SNMP version.

snmpInBadCommunityNames {snmp 4}
Counter Read-Only

The number of SNMP PDUs received by the unit which had an unrecognized SNMP community name.

snmpInBadCommunityUses {snmp 5}
Counter Read-Only

The number of SNMP PDUs received by the unit which had an authentication failure.

snmpInASNParseErrs {snmp 6}
Counter Read-Only

The number of SNMP PDUs received by the unit which had an ASN.1 parsing error while being decoded by the unit.

snmpInBadTypes {snmp 7}
Counter Read-Only

All GetResponse PDUs indicate a noSuchName ErrorStatus, since this variable is no longer used.

snmpInTooBigs {snmp 8}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInNoSuchNames {snmp 9}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInBadValues {snmp 10}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInReadOnlys {snmp 11}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInGenErrs {snmp 12}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInTotalReqVars {snmp 13}
Counter Read-Only

The total number of MIB objects which have been successfully retrieved by the unit as a result of SNMP GetRequest or GetNext PDUs.

snmpInTotalSetVars {snmp 14}
Counter Read-Only

The total number of MIB objects which have been successfully altered by the unit as a result of SNMP SetRequest PDUs.

snmpInGetRequests {snmp 15}
Counter Read-Only

The total number of SNMP GetRequest PDUs received by the unit, which have been processed with no errors.

snmpInGetNexts {snmp 16}
Counter Read-Only

The total number of SNMP GetNext PDUs received by the unit, which have been processed with no errors.

snmpInSetRequests {snmp 17}
Counter Read-Only

The total number of SNMP SetRequest PDUs received by the unit, which have been processed with no errors.

snmpInGetResponses {snmp 18}
Counter Read-Only

Always zero, since the unit ignores all SNMP response PDUs.

snmpInTraps {snmp 19}
Counter Read-Only

Always zero, since the unit ignores all SNMP Trap PDUs.

snmpOutTooBigs {snmp 20}
Counter Read-Only

The total number of SNMP PDUs created by the unit, with a value of “tooBig” in the PDU’s “ErrorStatus”.

snmpOutNoSuchNames {snmp 21}
Counter Read-Only

The total number of SNMP PDUs created by the unit, with a value of “noSuchName” in the PDU’s “ErrorStatus”.

snmpOutBadValues {snmp 22}
Counter Read-Only

The total number of SNMP PDUs created by the unit, with a value of “badValue” in the PDU’s “ErrorStatus”.

snmpOutReadOnlys {snmp 23}
Counter Read-Only

All GetResponse PDUs indicate a noSuchName ErrorStatus, since this variable is no longer used.

snmpOutGenErrs {snmp 24}
Counter Read-Only

The total number of SNMP PDUs created by the unit, with a value of “genErr” in the PDU’s “ErrorStatus”.

snmpOutGetRequests {snmp 25}
Counter Read-Only

Always zero, since the unit never creates any SNMP request PDUs.

snmpOutGetNexts {snmp 26}
Counter Read-Only

Always zero, since the unit never creates any SNMP request PDUs.

snmpOutSetRequests {snmp 27}
Counter Read-Only

Always zero, since the unit never creates any SNMP request PDUs.

snmpOutGetResponses {snmp 28}
Counter Read-Only

The total number of SNMP GetResponse PDUs created by the unit.

snmpOutTraps {snmp 29}
Counter Read-Only

The total number of SNMP Trap PDUs created by the unit.

snmpEnableAuthenTraps {snmp 30}
Integer Read-Write

Whether authentication failures should cause the unit to generate authentication-failure Trap PDUs. Values include:

- enabled (1) - generate traps
- disabled (2) - do not generate traps

CHAPTER 3
ETHERNET MIB

The unit supports the Ethernet MIB as defined in *Definitions of Managed Objects for the Ethernet-like Interface Types*, RFC 1284 (J. Cook, editor), dated December 1991.

The MIB tree prefix for reaching the Ethernet group is:

```
1.3.6.1.2.1.10.
```

transmission {mib-2 10}

3.1 GENERIC ETHERNET-LIKE GROUP

dot3 {transmission 7}

dot3Table {dot3 1}
Not-Accessible

A list of interface entries; one per Ethernet port. The entry instances correspond to the unit's port numbers. For example, if the unit's 2nd port is Ethernet, then both {ifindex 2} and {dot3Index 2} contain the value "2". If the unit's 2nd port is not Ethernet, then referencing {dot3Index 2} should return noSuchName.

dot3Entry {dot3Table 1}
Not-Accessible

A set of operational attributes for an Ethernet entry. The individual components are described below.

dot3Index {dot3Entry 1}
Integer Read-Only

The unit's Ethernet port number.

dot3InitializeMAC {dot3Entry 2}
Integer Read-Write

Whether the port has been initialized. Initializing a port automatically resets all of the port's statistics, and enables transmit and receive (the receiver will be in promiscuous mode). The value of dot3InitializeMAC is not saved in the event that the unit reboots. Values include:

- initialized (1)
- uninitialized (2)

dot3MACSubLayerStatus {dot3Entry 3}
Integer Read-Write

Treated identically to dot3InitializeMAC. Values include:

- enabled (1)
- disabled (2)

dot3MulticastReceiveStatus {dot3Entry 4}
Integer Read-Write

Whether the port is able to receive multicasts. This is always enabled when the port is enabled, and always disabled when the port is disabled. Values include:

- enabled (1)
- disabled (2)

dot3TxEnabled {dot3Entry 5}
Integer Read-Write

Whether the port is able to transmit. This is always enabled when the port is enabled, and always disabled when the port is disabled. Values include:

- true (1)
- false (2)

dot3TestTdrValue {dot3Entry 6}
Gauge Read-Only

Always zero, since the results of TDR tests are not available.

3.2 ETHERNET-LIKE STATISTICS GROUP

The dot3StatsTable is specified in a unique manner. That is, the RFC specifies that value of 0 should be returned for counters that are not implemented.

dot3StatsTable {dot3 2}
Not-Accessible

A list of interface entries; one per Ethernet port, which correspond to the dot3Table entries.

dot3StatsEntry {dot3StatsTable 1}
Not-Accessible

A set of statistics for an Ethernet entry. The individual components are described below.

dot3StatsIndex {dot3StatsEntry 1}
Integer Read-Only

The unit's Ethernet port number.

dot3StatsAlignmentErrors {dot3StatsEntry 2}
Counter Read-Only

Number of received packets with frame alignment errors since the port was last enabled.

dot3StatsFCSErrors {dot3StatsEntry 3}
Counter Read-Only

Number of received packets with FCS errors (and without frame alignment errors) since the port was last enabled.

dot3StatsSingleCollisionFrames {dot3StatsEntry 4}
Counter Read-Only

The total number of collisions, since the port was last enabled, which may or may not reflect the number of frames that experienced a single collision.

dot3StatsMultipleCollisionFrames {dot3StatsEntry 5}
Counter Read-Only

The number of successfully transmitted frames that experienced more than one collision.

dot3StatsSQETestErrors {dot3StatsEntry 6}
Counter Read-Only

The number of times an SQE failure (i.e., heartbeat lost) occurred, since the port was last enabled. This counter is only incremented when heartbeat is lost, if the port has been configured to expect heartbeat (see sifExpectSQE). Increments to this counter correspond to one or more packets experiencing SQE failure. In other words, a single increment to this counter may indicate that many packet transmissions experienced SQE failure.

dot3StatsDeferredTransmissions {dot3StatsEntry 7}
Counter Read-Only

The number of frames for which the first transmission attempt was delayed because the medium was busy. Zero is always returned.

dot3StatsLateCollisions {dot3StatsEntry 8}
Counter Read-Only

The number of times that a late collision was detected. Zero is always returned.

dot3StatsExcessiveCollisions {dot3StatsEntry 9}
Counter Read-Only

The number of times a frame transmission attempt experienced excessive collisions, since the port was last enabled.

dot3StatsInternalMACTransmitErrors {dot3StatsEntry 10}
Counter Read-Only

The number of times a frame transmission attempt experienced an error other than dot3StatsSQETestErrors, dot3StatsExcessiveCollisions, or dot3StatsCarrierSenseErrors, since the port was last enabled.

dot3StatsCarrierSenseErrors {dot3StatsEntry 11}
Counter Read-Only

The number of times that a carrier sense condition was lost during a transmission attempt, since the port was last enabled.

dot3StatsExcessiveDeferrals {dot3StatsEntry 12}
Counter Read-Only

The number of times that transmission was deferred for an excessive period of time. Zero is always returned.

dot3StatsFrameTooLongs {dot3StatsEntry 13}
Counter Read-Only

The number of times a received packet was too long, since the port was last enabled.

dot3StatsInRangeLengthErrors {dot3StatsEntry 14}
Counter Read-Only

The number of received frames with an incorrect LLC data size. Zero is always returned.

dot3StatsOutOfRangeLengthErrors {dot3StatsEntry 15}
Counter Read-Only

The number of received frames with an LLC data size that is too large (not including frames that have an Ethernet frame type field). Zero is always returned.

dot3StatsInternalMACReceiveErrors {dot3StatsEntry 16}
Counter Read-Only

The number of times a frame reception failed due to an error condition not already accounted for, since the port was last enabled.

3.3 ETHERNET-LIKE COLLISION STATISTICS GROUP

dot3CollTable {dot3 5}

This group is minimally supported (i.e., the unit returns a GetResponse with a noSuchName ErrorStatus).

3.4 ETHERNET-LIKE TESTS GROUP

dot3Tests {dot3 6}

There are no MIB variables in this group.

3.5 ETHERNET-LIKE ERRORS GROUP

dot3Errors {dot3 7}

There are no MIB variables in this group.

3.6 ETHERNET-LIKE CHIPSETS GROUP

dot3ChipSets {dot3 8}

There are no MIB variables in this group.

CHAPTER 4
TOKEN RING MIB

The unit supports the Token Ring MIB as defined in *IEEE 802.5 Token Ring MIB*, RFC 1231 (K. McCloghrie, editor) dated May 1991. Note that *Reassignment of Experimental MIBs to Standard MIBs*, RFC 1239 (J. Reynolds, editor), dated June 1991, has reassigned the Token Ring MIB from the experimental namespace to the transmission namespace.

The MIB branch for reaching the Token Ring group is:

```
1.3.6.1.2.1.10.
```

4.1 TOKEN RING GROUP

dot5 {transmission 9}

4.1.1 Token Ring Interface Table

dot5Table {dot5 1}
Not-Accessible

This table contains Token Ring interface parameters and state variables, on entry per 802.5 interface.

The entry instances correspond to the unit's port numbers. For example, if the unit's 2nd port is 802.5, then both {ifIndex2} and {dot5IfIndex 2} contain the value "2". If the unit's 2nd port is not 802.5, then referencing {dot5IfIndex 2} should return noSuchName.

dot5Entry {dot5Table 1}
Not-Accessible

A list of Token Ring status and parameter values for an 802.5 interface.

dot5ifIndex {dot5Entry 1}
Integer Read-Only

The value of this object identifies the 802.5 interface for which this entry contains management information. The value of this object for a particular interface has the same value as the ifIndex object for the same interface.

dot5Commands {dot5Entry 2}
Integer Read-Write

When this object is set to the value of open (2), the station should go into the open state. The progress and success of the open is given by the values of the objects dot5RingState and dot5RingOpenStatus. When this object is set to the value of reset(3), then the station should do a reset. On a reset, all MIB counters should retain their values, if possible. Other side affects are dependent on the hardware chip set. When this object is set to the value of close(4), the station should go into the stopped state by removing itself from the ring. Setting this object to a value of no-op(1) has no effect. When read, this object always has a value of no-op(1).

dot5RingStatus {dot5Entry 3}
Integer Read-Only

The current interface status which can be used to diagnose fluctuating problems that can occur on token rings, after a station has successfully been added to the ring. Before an open is completed, this object has the value for the “no status” condition. The dot5RingState and dot5RingOpenStatus objects provide for debugging problems when the station can not even enter the ring. The object’s value is a sum of values, one for each currently applicable condition. The following values are defined for various conditions:

- 0 = No Problems detected
- 32 = Ring Recovery
- 64 = Single Station
- 256 = Remove Received
- 512 = reserved
- 1024 = Auto-Removal Error
- 2048 = Lobe Wire Fault
- 4096 = Transmit Beacon
- 8192 = Soft Error
- 16348 = Hard Error
- 32768 = Signal Loss
- 131072 = no status, open not completed

dot5RingState {dot5Entry 4}
Integer Read-Only

The current interface state with respect to entering or leaving the ring.

Values include:

- opened (1)
- closed (2)
- opening (3)
- closing (4)
- openFailure (5)
- ringFailure (6)

dot5RingOpenStatus {dot5Entry 5}
Integer Read-Only

This object indicates the success, or the reason for failure, of the station's most recent attempt to enter the ring.

Values include:

- noOpen (1) - no open attempted
- badParam (2)
- lobeFailed (3)

dot5RingSpeed {dot5 Entry 6}
Integer Read-Write

The ring's bandwidth.

dot5UpStream {dot5Entry 7}
MacAddress Read-Only

The Mac address of the up stream neighbor station in the ring.

dot5ActMonParticipate {dot5Entry 8}
Integer Read-Write

If this object has a value of true(1) then this interface will participate in the active monitor selection process. If the value is false(2) then it will not. Setting this object might not have an effect until the next time the interface is opened.

dot5Functional {dot5Entry 9}
MacAddress Read-Write

The bit mask of all Token Ring functional addresses for which this interface will accept frames.

4.1.2 Token Ring Statistics Table

dot5StatsTable {dot5 2}
Not-Accessible

A table containing Token Ring statistics, one entry per 802.5 interface. All the statistics are defined using the syntax Counter as 32 bit wrap around counters.

The entry instances correspond to the unit's port numbers. For example, if the unit's 2nd port is 802.5, then both {ifIndex 2} and {dot5StatsIfIndex 2} contain the value "2". If the unit's 2nd port is not 802.5, then referencing {dot5StatsIfIndex 2} should return noSuchName.

dot5StatsEntry {dot5StatsTable 1}
Not-Accessible

An entry contains the 802.5 statistics for a particular interface.

dot5StatsIfIndex {dot5StatsEntry 1}
Integer Read-Only

The value of this object identifies the 802.5 interface for which this entry contains management information. The value of this object for a particular interface has the same value as the ifIndex object for the same interface.

dot5StatsLineErrors {dot5StatsEntry 2}
Counter Read-Only

This counter is incremented when a frame or token is copied or repeated by a station, the E bit is zero in the frame or token and one of the following conditions exists: 1) there is a non-data bit (J or K bit) between the SD and the ED of the frame or token, or 2) there is an FCS error in the frame.

dot5StatusBurstErrors {dot5StatsEntry 3}
Counter Read-Only

This counter is incremented when a station detects the absence of transitions for five half-bit timers (burst-five errors).

dot5StatsACErrors {dot5StatsEntry 4}
Counter Read-Only

This counter is incremented when a station receives an AMP or SMP frame in which A is equal to C is equal to 0, and then receives another SMP frame with A is equal to C is equal to 0 without first receiving an AMP frame. It denotes a station that cannot set the AC bits properly.

dot5StatsAbortTransErrors {dot5StatsEntry 5}
Counter Read-Only

This counter is incremented when a station transmits an abort delimiter while transmitting.

dot5StatsInternalErrors {dot5StatsEntry 6}
Counter Read-Only

This counter is incremented when a station recognizes an internal error.

dot5StatsLostFrameErrors {dot5StatsEntry 7}
Counter Read-Only

This counter is incremented when a station is transmitting and its TRR timer expires. This condition denotes a condition where a transmitting station in strip mode does not receive the trailer of the frame before the TRR timer goes off.

dot5StatsReceiveCongestions {dot5StatsEntry 8}
Counter Read-Only

This counter is incremented when a station recognizes a frame addressed to its specific address, but has no available buffer space indicating that the station is congested.

dot5StatsFrameCopiedErrors {dot5StatsEntry 9}
Counter Read-Only

This counter is incremented when a station recognizes a frame addressed to its specific address and detects that the FS field A bits are set to 1 indicating a possible line hit or duplicate address.

dot5StatsTokenErrors {dot5StatsEntry 10}
Counter Read-Only

This counter is incremented when a station acting as the active monitor recognizes an error condition that needs a token transmitted.

dot5StatsSoftErrors {dot5StatsEntry 11}
Counter Read-Only

The number of Soft Errors the interface has detected. It directly corresponds to the number of Report Error MAC frames that this interface has transmitted. Soft Errors are those which are recoverable by the MAC layer protocols.

dot5StatsHardErrors {dot5StatsEntry 12}
Counter Read-Only

The number of times this interface has detected an immediately recoverable fatal error. It denotes the number of times this interface is either transmitting or receiving beacon MAC frames.

dot5StatsSignalLoss {dot5StatsEntry 13}
Counter Read-Only

The number of times this interface has detected the loss of signal condition from the ring.

dot5StatsTransmitBeacons {dot5StatsEntry 14}
Counter Read-Only

The number of times this interface has transmitted a beacon frame.

dot5StatsRecoverys {dot5StatsEntry 15}
Counter Read-Only

The number of Claim Token MAC frames received or transmitted after the interface has received a Ring Purge MAC frame. This counter signifies the number of times the ring has been purged and is being recovered back into a normal operating state.

dot5StatsLobeWires {dot5StatsEntry 16}
Counter Read-Only

The number of times the interface has detected an open or short circuit in the lobe data path. The adapter will be closed and dot5RingState will signify this condition.

dot5StatsRemoves {dot5StatsEntry 17}
Counter Read-Only

The number of times the interface has received a Remove Ring Stations MAC frame request. When this frame is received the interface will enter the close state and dot5RingState will signify this condition.

dot5StatsSingles {dot5StatsEntry 18}
Counter Read-Only

The number of times the interface has sensed that it is the only station on the ring. This will happen if the interface is the first one up on a ring, or if there is a hardware problem.

dot5StatsFreqErrors {dot5StatsEntry 19}
Counter Read-Only

The number of times the interface has detected that the frequency of the incoming signal differs from the expected frequency by more than that specified by the IEEE 802.5 standard.

4.1.3 Token Ring Timer Table

dot5TimerTable {dot5 3}

This entire group is just minimally supported (i.e., the unit returns a GetResponse with a noSuchName ErrorStatus).

4.1.4 Token Ring Tests

dot5Tests {dot5 4}

There are no MIB variables in this group.

4.1.5 Token Ring Chipsets

dot5ChipSets {dot5 5}

There are no MIB variables in this group.

CHAPTER 5
FDDI MIB

The ATX supports the FDDI MIB as defined in *FDDI Management Information Base*, RFC 1512 (Jeffery D. Case, editor) dated September 1993. The FDDI MIB is divided into groups of parameters. The individual groups are described in the subsections below. You may also need to refer to the RFC, since this document paraphrases the standard MIB in order to provide device specific descriptions.

The access types of some of the parameters have been extended from *Read-Only* to *Read-Write*, indicatd by an asterisk (*); other parameters have been changed from *Read-Write* to *Read-Only*, indicated by a number sign (#). The access type of all other parameters is as indicated in the standard MIB.

The value of fdditime is measured in number of nonseconds; in previous releases it was measured in number of 80 nanosecond units.

The MIB branch for reaching the SMT group is:

```
1.3.6.1.2.1.10.
```

5.1 SMT GROUP

fddi {transmission 15}

This group supports SMT version 7.3. Version 6.2 was supported in release 2.3 of ATX system software however, it has been deprecated and thus is not included in this manual.

fddimib {fddi 73}

fddimibSMT {fddimib 1}

fddimibSMTNumber {fddimibSMT 1}
Integer Read-Only

The number of FDDI ports configured in Configuration EEPROM.

5.1.1 SMT Table

fddimibSMTTable {fddimibSMT 2}
Not-Accessible

A list of interface entries; one per FDDI port. The entries are indexed by fddimibSMTIndex.

fddimibSMTEntry {fddimibSMTTable 1}
Not-Accessible

A set of operational attributed for an FDDI port. The individual components are described below.

fddimibSMTIndex {fddimibSMTEntry 1}
Integer Read-Only

A unique value for each SMT. The values do not correspond to the unit's port numbers. For example, parameter instance {fddimibSMTIndex 1} corresponds to the unit's first FDDI port, which cannot be port number 1, since port number 1 is the UART.

fddimibSMTStationID {fddimibSMTEntry 2}
Octet String Read-Write*

Used to uniquely identify an FDDI station (corresponds to ANSI {fddiSMT 11}). Setting the lower six bytes to zero will cause the unit to automatically use the port's MAC address in the lower six bytes.

fddimibSMTOpVersionID {fddimibSMTEntry 3}
Integer Read-Only

The version that this station is using for its operation (corresponds to ANSI {fddiSMT 13}, refer to ANSI 7.1.2.2). The value of this variable is 2 for this SMT revision.

fddimibSMTHiVersionID {fddimibSMTEntry 4}
Integer Read-Only

The highest version of SMT that this station supports (corresponds to ANSI {fddiSMT 14}, refer to ANSI 7.1.2.2).

fddimibSMTLoVersionID {fddimibSMTEntry 5}
Integer Read-Only

The lowest version of SMT that this station supports (corresponds to ANSI {fddiSMT 15}, refer to ANSI 7.1.2.2).

fddimibSMTUserData {fddimibSMTEntry 6}
Octet String Read-Write

This variable contains 32 octets of user defined information (corresponds to ANSI {fddiSMT 17}). The information shall be an ASCII string.

fddimibSMTMIBVersionId {fddimibSMTEntry 7}
Integer Read-Only

The version of the FDDI MIB of this station (corresponds to ANSI {fddiSMT 18}). The value of this variable is 1 for this SMT revision.

fddimibSMTMACCts {fddimibSMTEntry 8}
Integer Read-Only

The number of MACs in this station or concentrator (corresponds to ANSI {fddiSMT 21}).

fddimibSMTNonMasterCts {fddimibSMTEntry 9}
Integer Read-Only

The value of this variable is the number of A, B, and S ports in this station or concentrator (corresponds to ANSI {fddiSMT 22}).

fddimibSMTMasterCts {fddimibSMTEntry 10}
Integer Read-Only

The number of M Ports in a node. If the node is not a concentrator, the value of the variable is zero (corresponds to ANSI {fddiSMT 23}).

fddimibSMTAvailablePaths {fddimibSMTEntry 11}
Integer Read-Only

A value that indicates the PATH types available in the station (corresponds to ANSI {fddiSMT 24}).

fddimibSMTConfigCapabilities {fddimibSMTEntry 12}
Integer Read-Only

A value that indicates the configuration capabilities of a node (corresponds to ANSI {fddiSMT 25}). This is always 0.

fddimibSMTConfigPolicy {fddimibSMTEntry 13}
Integer Read-Only#

A value that indicates the configuration policies currently desired in a node (corresponds to ANSI {fddiSMT 26}). This is always 0.

fddimibSMTConnectionPolicy {fddimibSMTEntry 14}
Integer Read-Write

The connection restrictions that are currently being enforced by the port (corresponds to ANSI {fddiSMT 27}), which are any combinations of the following bits:

```
CONN_REJECT_AA = 0x00000001
CONN_REJECT_AB = 0x00000002
CONN_REJECT_AS = 0x00000004
CONN_REJECT_AM = 0x00000008
CONN_REJECT_BA = 0x00000010
CONN_REJECT_BB = 0x00000020
CONN_REJECT_BS = 0x00000040
CONN_REJECT_BM = 0x00000080
CONN_REJECT_SA = 0x00000100
CONN_REJECT_SB = 0x00000200
CONN_REJECT_SS = 0x00000400
CONN_REJECT_SM = 0x00000800
CONN_REJECT_MA = 0x00001000
CONN_REJECT_MB = 0x00002000
CONN_REJECT_MS = 0x00004000
CONN_REJECT_MM = 0x00008000
```

fddimibSMTTNotify {fddimibSMTEntry 15}
Integer Read-Write

The timer, expressed in seconds, used in the Neighbor Notification protocol (corresponds to ANSI {fddiSMT 29}). It has a range of 2 seconds to 30 seconds, and its default value is 30 seconds (refer to ANSI SMT 8.2).

fddimibSMTStatRptPolicy {fddimibSMTEntry 16}
Integer Read-Write

If true (1), indicates that the node will generate Status Reporting Frames for its implemented events and conditions (corresponds to ANSI {fddiSMT 30}). It has an initial value of true. This variable determines the value of the SR_Enable Flag (refer to ANSI SMT 8.3.2.1).

fddimibSMTTraceMaxExpiration {fddimibSMTEntry 17}
Integer Read-Write

Reference Trace_Max (corresponds to ANSI {fddiSMT 31}), (refer to ANSI SMT 9.4.4.2.2).

fddimibSMTBypassPresent {fddimibSMTEntry 18}
Integer Read-Only

A flag indicating if the station has a bypass on its AB port pair (corresponds to ANSI {fddiSMT 34}).

fddimibSMTECMState {fddimibSMTEntry 19}
Integer Read-Only

Indicates the current state of the ECM state machine (corresponds to ANSI {fddiSMT 41}, refer to ANSI SMT 9.5.2).

fddimibSMTCFMState {fddimibSMTEntry 20}
Integer Read-Only

The attachment configuration for the station or concentrator (corresponds to ANSI {fddiSMT 42}, refer to ANSI SMT 9.7.2.2).

fddimibSMTRemoteDisconnectFlag {fddimibSMTEntry 21}
Integer Read-Only

Whether the port has been intentionally disconnected (corresponds to ANSI {fddiSMT 44}). The NMS may cause an intentional disconnect, or, the unit may be configured to not use the FDDI port.

fddimibSMTStationStatus {fddimibSMTEntry 22}
Integer Read-Only

The current status of the primary and secondary paths within this station (corresponds to ANSI {fddiSMT 45}).

fddimibSMTPeerWrapFlag {fddimibSMTEntry 23}
Integer Read-Only

This variable assumes the value of the PeerWrapFlag in CFM (corresponds to ANSI {fddiSMT 46}, refer to ANSI SMT 9.7.2.4.4).

fddimibSMTTimeStamp {fddimibSMTEntry 24}
Integer Read-Only

This variable assumes the value of TimeStamp, in the unit of milliseconds (corresponds to ANSI {fddiSMT 51}, refer to ANSI SMT 8.3.2.1).

fddimibSMTTransitionTimeStamp {fddimibSMTEntry 25}
Integer Read-Only

This variable assumes the value of the TransitionTimeStamp, in the unit of milliseconds (corresponds to ANSI {fddiSMT 52}, refer to ANSI SMT 8.3.2.1).

fddimibSMTStationAction {fddimibSMTEntry 26}
Integer Read-Write

An SMT action to be performed (corresponds to ANSI {fddiSMT 60}. Only connect (2) and disconnect (3) are supported. The value of fddimibSMTStationAction is not saved in the event that the unit reboots.

5.2 MAC GROUP

fddimibMAC {fddimib 2}

FddimibMACNumber {fddimibMAC 1}
Integer Read-Only

The number of FDDI port configured in Configuration EEPROM. Note that fddimibMACNumber and fddimibSMTNumber always have identical values in the unit.

5.2.1 MAC Table

fddimibMACTable {fddimibMAC 2}
Not-Accessible

A list of interface entries; one per FDDI port. The entries are indexed by fddimibMACSMTIndex and fddimibMACIndex.

fddimibMACEntry {fddimibMACTable 1}
Not-Accessible

A set of operational attributes for an FDDI port. The individual components are described below.

fddimibMACSMTIndex {fddimibMACEntry 1}
Integer Read-Only

The value of the SMT index associated with this MAC. The values do not correspond to the unit's port numbers. For example, parameter instance {fddimibMACSMTIndex 1} corresponds to the unit's first FDDI port, which cannot be port number 1, since port number 1 is the UART.

fddimibMACIndex {fddimibMACEntry 2}
Integer Read-Only

Index variable for uniquely identifying the MAC object instances, which is the same as the corresponding resource index in SMT (corresponds to ANSI {fddiMAC 34}). This is always 3.

fddimibMACIfIndex {fddimibMACEntry 3}
Integer Read-Only

The unit's port number, which can be used to "point to" the corresponding ifEntry. This number always matches the corresponding instance of the ifEntry, For example, if the unit has a single FDDI port which is the unit's port "n", then {fddimibMACIfIndex 1} contains "n", and the "nth" instance of ifEntry contains additional information concerning the port.

fddimibMACFrameStatusFunctions {fddimibMACEntry 4}
Integer Read-Only

Indicates the MAC's optional Frame Status processing functions (corresponds to ANSI {fddiMAC 11}). This is always 6 (fs-setting and fs-clearing).

fddimibMACTMaxCapability {fddimibMACEntry 5}
FDDITime Read-Only

Indicates the maximum time value (in nanoseconds) of fddiMACTMax that this MAC can support (corresponds to ANSI {fddiMAC 13}).

fddimibMACTVXCapability {fddimibMACEntry 6}
FDDITime Read-Only

Indicates the maximum time value (in nanoseconds) of fddiMACTvxValue that this MAC can support (corresponds to ANSI {fddiMAC 14}).

fddimibMACAvailablePaths {fddimibMACEntry 7}
Integer Read-Only

Indicates the paths available for this MAC (corresponds to ANSI {fddiMAC 22}, refer to ANSI SMT 9.7.7). The value is always 3 (primary and secondary).

fddimibMACCurrentPath {fddimibMACEntry 8}
Integer Read-Only

Indicates the path into which this MAC is currently inserted (corresponds to ANSI {fddiMAC 23}, refer to ANSI SMT 9.7.7).

fddimibMACUpstreamNbr {fddimibMACEntry 9}
Octet String Read-Only

The MAC's upstream neighbor's long individual MAC address (corresponds to ANSI {fddiMAC 24}. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2).

fddimibMACDownstreamNbr {fddimibMACEntry 10}
Octet String Read-Only

The MAC's downstream neighbor's long individual MAC address (corresponds to ANSI {fddiMAC 25}. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2).

fddimibMACOldUpstreamNbr {fddimibMACEntry 11}
Octet String Read-Only

The previous value of the MAC's upstream neighbor's long individual MAC address (corresponds to ANSI {fddiMAC 26}. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2).

fddimibMACOldDownstreamNbr {fddimibMACEntry 12}
Octet String Read-Only

The previous value of the MAC's downstream neighbor's long individual MAC address (corresponds to ANSI {fddiMAC 27}. It has an initial value of the SMT-Unknown-MAC Address and is only modified as specified by the Neighbor Information Frame protocol (refer to ANSI SMT 7.2.1 and 8.2).

fddimibMACDupAddressTest {fddimibMACEntry 13}
Integer Read-Only

The Duplicate Address Test flag, Dup_Addr_Test (corresponds to ANSI {fddiMAC 29} refer to ANSI 8.2).

fddimibMACRequestedPaths {fddimibMACEntry 14}
Integer Read-Only#

List of permitted Paths which specifies the Path (s) into which the MAC may be inserted (corresponds to ANSI {fddiMAC 32} refer to ANSI SMT 9.7). Only default setting as defined in SMT 9.7 is supported.

fddimibMACDownstreamPORTType {fddimibMACEntry 15}
Integer Read-Only

Indicates the PC-Type of the first port that is downstream of this MAC (corresponds to ANSI {fddiMAC 33}).

fddimibMACSMTAddress {fddimibMACEntry 16}
Octet String Read-Only

The 48-bit individual address of the MAC used for SMT frames (corresponds to ANSI {fddiMAC 41}).

fddimibMACTReq {fddimibMACEntry 17}
FddiTime Read-Only

This variable is the T_Req_value passed to the MAC (corresponds to ANSI {fddiMAC 51}).

fddimibMACTNeg {fddimibMACEntry 18}
FddiTime Read-Only

The T_Neg value determined by this MAC (corresponds to ANSI {fddiMAC 52}).

fddimibMACTMax {fddimibMACEntry 19}
FddiTime Read-Only

This variable is the T_Max_value passed to the MAC (corresponds to ANSI {fddiMAC 53}).

fddimibMACTvxValue {fddimibMACEntry 20}
FddiTime Read-Only

This variable is the TVX_value passed to the MAC (corresponds to ANSI {fddiMAC 53}).

fddimibMACFrameCts {fddimibMACEntry 21}
Counter Read-Write*

A count of the number of frames received by this MAC (corresponds to ANSI {fddiMAC 71}, refer to ANSI MAC 7.5.1). Any attempt to write this parameter will cause the MAC's Frame_Ct, Copied_Ct, Transmit_Ct, Error_Ct, and Lost_Ct to be zeroed.

fddimibMACCopiedCts {fddimibMACEntry 22}
Counter Read-Write*

A count that should as closely as possible match the number of frames addressed to a bit set and successfully copied into the station's receive buffers (C bit set) by the MAC (corresponds to ANSI {fddiMAC 72}, refer to ANSI MAC 7.5). Note that this count does not include MAC frames. Any attempt to write this parameter will cause the MAC's Frame_Ct, Copied_Ct, Transmit_Ct, Error_Ct, and Lost_CT to be zeroed.

fddimibMACTransmitCts {fddimibMACEntry 23}
Counter Read-Write*

A count that should as closely as possible match the number of frames transmitted by this MAC (corresponds to ANSI {fddiMAC 73}, refer to ANSI MAC 7.5). Note that this count does not include MAC frames. Any attempt to write this parameter will cause the MAC's Frame_Ct, Copied_Ct, Transmit_Ct, Error_Ct, and Lost_CT to be zeroed.

fddimibMACErrorCts {fddimibMACEntry 24}
Counter Read-Write*

A count of the number of frames that were detected in error by this MAC that had not been detected in error by another MAC (corresponds to ANSI {fddiMAC 81}, refer to ANSI MAC 7.5.2). Note that this count does not include MAC frames. Any attempt to write this parameter will cause the MAC's Frame_Ct, Copied_Ct, Transmit_Ct, Error_Ct, and Lost_CT to be zeroed.

fddimibMACLostCts {fddimibMACEntry 25}
Counter Read-Write*

A count of the number of instances that this MAC detected a format error during frame reception such that the frame was stripped (corresponds to ANSI {fddiMAC 82}, refer to ANSI MAC 7.5.3). Any attempt to write this parameter will cause the MAC's Frame_Ct, Copied_Ct, Transmit_Ct, Error_Ct, and Lost_CT to be zeroed.

fddimibMACFrameErrorThreshold {fddimibMACEntry 26}
Integer Read-Write

A threshold (0..65536) for determining when a MAC Condition Report (corresponds to ANSI {fddiMAC 95}, refer to ANSI MAC 8.3.1.1) shall be generated.

fddimibMACFrameErrorRatio {fddimibMACEntry 27}
Integer Read-Only

The actual frame error ratio (corresponds to ANSI {fddiMAC 96}).

fddimibMACRMTState {fddimibMACEntry 28}
Integer Read-Only

Indicates the current state of the RMT State Machine (corresponds to ANSI {fddiMAC 111}, refer to ANSI MAC 10.3.2).

fddimibMACDaFlag {fddimibMACEntry 29}
Integer Read-Only

The RMT flag Duplicate Address Flag, DA_Flag (corresponds to ANSI {fddiMAC 112}, refer to ANSI MAC 10.2.1.2).

fddimibMACUnaDaFlag {fddimibMACEntry 30}
Integer Read-Only

A flag, UNDA_Flag (corresponds to ANSI {fddiMAC 113}, refer to ANSI MAC 8.2.2.1), set when the upstream neighbor reports a duplicate address condition. Cleared when the condition clears.

fddimibMACFrameErrorFlag {fddimibMACEntry 31}
Integer Read-Only

Indicates the MAC Frame Error Condition is present when set. Cleared when the condition clears and on station initialization (corresponds to ANSI {fddiMAC 114}).

fddimibMACMAUnitdataAvailable {fddimibMACEntry 32}
Integer Read-Only

This variable shall take on the value of the MAC_Avail flag defined in RMT (corresponds to ANSI {fddiMAC 116}).

fddimibMACHardwarePresent {fddimibMACEntry 33}
Integer Read-Only

This variable indicates the presence of underlying hardware support for this MAC object (corresponds to ANSI {fddiMAC 117}). This is always true (1).

fddimibMACMAUnitdataEnable {fddimibMACEntry 34}
Integer Read-Write

This variable determines the value of the MA_UNITDATA_Enable flag in RMT (corresponds to ANSI {fddiMAC 118}.

5.3 MAC COUNTER GROUP

fddimibMACCounters {fddimib 3}

This is the Enhanced MAC Counters group. Implementation of this Group is optional, but systems claiming support must implement all variables in this group. This device does not support this Group.

5.4 PATH GROUP

fddimibPATH {fddimib 4}

fddimibPATHNumber {fddimibPATH 1}
Integer Read-Only

The total number of PATHs possible (across all FDDI ports) on this unit.

5.4.1 PATH Table

fddimibPATHTable {fddimibPATH 2}
Not-Accessible

A list of interface entries; one per each path. The entries are indexed by fddimibPATHSMTIndex and fddimibPATHIndex.

fddimibPATHEntry {fddimibPATHTable 1}
Not-Accessible

A set of operational attributes for a path. The individual components are described below.

fddimibPATHSMTIndex {fddimibPATHEntry 1}
Integer Read-Only

The value of the SMT index associated with this PATH.

fddimibPATHIndex {fddimibPATHEntry 2}
Integer Read-Only

Index variable for uniquely identifying the primary and secondary PATH object instances (corresponds to ANSI {fddiPATH 11}). This is 1 (for a primary path) or 2 (for a secondary path).

fddimibPATHTVXLowerBound {fddimibPATHEntry 3}
FddiTime Read-Write

Specifies the minimum time value of fddiMACTvxValue that shall be used by the MAC that is configured in this path (corresponds to ANSI {fddiPATH 21}). The operational value of fddiMACTvxValue is managed by setting this variable. Possible values range from 40960 (40.96 microseconds) to 5202000 (5.2 milliseconds).

fddimibPATHTMaxLowerBound {fddimibPATHEntry 4}
FddiTime Read-Write

Specifies the minimum time value of fddiMACTMax that shall be used by the MAC that is configured in this path (corresponds to ANSI {fddiPATH 22}). The operational value of fddiMACTMax is managed by setting this variable. Possible values range from 40960 (40.96 microseconds) to 167769600 (167,7696 milliseconds).

fddimibPATHMaxTReq {fddimibPATHEntry 5}
FddiTime Read-Write

Specifies the maximum time value of fddiMACTReq that shall be used by the MAC that is configured in this path. The operational value of fddiMACTReq is managed by setting this variable. Possible values are the same as fddimibPATHTmaxLowerBound.

5.4.2 PATH Config Table

fddimibPATHConfigTable {fddimibPATH 3}
Not-Accessible

A table of Path configuration entries. The entries are indexed by fddimibPATHConfigSMTIndex, fddimibPATHConfigPATHIndex and fddimibPATHConfigTokenOrder.

fddimibPATHConfigEntry {fddimibPATHConfigTable 1}
Non-Accessible

A set of operational attributes for a path configuration entry. The individual components are described below.

fddimibPATHConfigSMTIndex {fddimibPATHConfigEntry 1}
Integer Read-Only

The value of the SMT index associated with this configuration entry.

fddimibPATHConfigPATHIndex {fddimibPATHConfigEntry 2}
Integer Read-Only

The value of the PATH resource index associated with this configuration entry.

fddimibPATHConfigTokenOrder {fddimibPATHConfigEntry 3}
Integer Read-Only

An object associated with Token order for this entry. Thus if the token passes resources a, b, c, and d, in that order, then the value of this object for these resources would be 1, 2, 3, and 4, respectively.

fddimibPATHConfigResourceType {fddimibPATHConfigEntry 4}
Integer Read-Only

The type of resource associated with this configuration entry. The value is 2 (for a MAC) or 4 (for a PHY).

fddimibPATHConfigResourceIndex {fddimibPATHConfigEntry 5}
Integer Read-Only

The value of the SMT resource index used to refer to the instance of this MAC or Port resource.

fddimibPATHConfigCurrentPath {fddimibPATHConfigEntry 6}
Integer Read-Only

The current insertion status for this resource on this Path.

5.5 PORT GROUP

fddimibPORT {fddimib 5}

The FDDI PORT Group parameters are the following:

fddimibPORTNumber {fddimibPORT 1}
Integer Read-only

The total number of PHYS possible (across all FDDI ports) on this unit.

5.5.1 PORT Table

fddimibPORTTable {fddimibPORT 2}
Not-Accessible

A list of interface entries; one per each PHY. The entries are indexed by fddimibPORTSMTIndex and fddimibPORTIndex.

fddimibPORTEntry {fddimibPORTTable 1}
Not-Accessible

A set of operational attributes for a PHY entry. The individual components are described below.

fddimibPORTSMTIndex {fddimibPORTEntry 1}
Integer Read-Only

The value of the SMT index associated with this PORT.

fddimibPORTIndex {fddimibPORTEntry 2}
Integer Read-Only

A unique value for each PORT within a given SMT, which is the same as the corresponding resource index in SMT (corresponds to ANSI {fddiPORT 29}). This is 1 (for a PHY A) or 2 (for a PHY B).

fddimibPORTMyType {fddimibPORTEntry 3}
Integer Read-Only

The value of the PORT's PC_Type (corresponds to ANSI {fddiPORT 12}, refer to ANSI 9.4.1, and 9.6.3.2).

fddimibPORTNeighborType {fddimibPORTEntry 4}
Integer Read-Only

The type of the remote PORT as determined in PCM (corresponds to ANSI {fddiPORT 13}).

fddimibPORTConnectionPolicies {fddimibPORTEntry 5}
Integer Read-Only#

A value representing the PORT's connection policies desired in the node (corresponds to ANSI {fddiPORT 14}).

fddimibPORTMACIndicated {fddimibPORTEntry 6}
Integer Read-Only

The indications (T_Val (9), R_Val (9) in PC_Signalling, of the intent to place a MAC in the output token path to a PORT (corresponds to ANSI {fddiPORT 15}, refer to ANSI SMT 9.6.3.2).

fddimibPORTCurrentPath {fddimibPORTEntry 7}
Integer Read-Only

Indicates the Path (s) into which this PORT is currently inserted (corresponds to ANSI {fddiPORT 16}).

fddimibPORTRequestedPaths {fddimibPORTEntry 8}
Octet String Read-Only#

This variable is a list of permitted Paths where each list element defines the Port's permitted Paths (corresponds to ANSI {fddiPORT 17}). The first octet corresponds to "none", the second octet to "tree", and the third octet to "peer".

fddimibPORTMACPlacement {fddimibPORTEntry 9}
Integer Read-Only

Indicated the MAC, if any, whose transmit path exits the station via this PORT (corresponds to ANSI {fddiPORT 18}).

fddimibPORTAvailablePaths {fddimibPORTEntry 10}
Integer Read-Only

Indicates the Paths which are available to the Port (corresponds to ANSI {fddiPORT 19}). In the absence of faults, the A and B Ports will always have both the Primary and Secondary Paths available.

fddimibPORTPMDClass {fddimibPORTEntry 11}
Integer Read-Only

This variable indicates the type of PMD entity associated with this port (corresponds to ANSI {fddiPORT 22}).

fddimibPORTConnectionCapabilities {fddimibPORTEntry 12}
Integer Read-Only

A value that indicates the connection capabilities of the port (corresponds to ANSI {fddiPORT 23}).

fddimibPORTBSFlag {fddimibPORTEntry 13}
Integer Read-Only

This variable assumes the value of the BS_Flag (corresponds to ANSI {fddiPORT 33}, refer to ANSI SMT 9.4.3.3).

fddimibPORTLCTFailCts {fddimibPORTEntry 14}
Counter Read-Only

The count of the consecutive times the link confidence test (LCT) has failed during connection management (corresponds to ANSI {fddiPORT 42}, refer to ANSI 9.4.1).

fddimibPORTLerEstimate {fddimibPORTEntry 15}
Integer Read-Only

A long term average link error rate (corresponds to ANSI {fddiPORT 51}, refer to ANSI SMT9.4.7.5).

fddimibPORTLemRejectCts {fddimibPORTEntry 16}
Counter Read-Only

A link error monitoring count of the times that a link has been rejected (corresponds to ANSI {fddiPORT 52}).

fddimibPORTLemCts {fddimibPORTEntry 17}
Integer Read-Only

The aggregate link error monitor error count, set to zero only on station initialization (corresponds to ANSI {fddiPORT 53}).

fddimibPORTLerCutoff {fddimibPORTEntry 18}
Integer Read-Write

The link error rate estimate at which a link connection will be broken (corresponds to ANSI {fddiPORT 58}).

fddimibPORTLerAlarm {fddimibPORTEntry 19}
Integer Read-Write

The link error rate estimate at which a link connection will generate an alarm (corresponds to ANSI {fddiPORT 59}).

fddimibPORTConnectState {fddimibPORTEntry 20}
Integer Read-Only

An indication of the connect state of the PORT and is equal to the value of Connect_State (corresponds to ANSI {fddiPORT 61}, refer to ANSI 9.4.1).

fddimibPORTPCMState {fddimibPORTEntry 21}
Integer Read-Only

The state of the Port's PCM state machine (corresponds to ANSI {fddiPORT 62}, refer to ANSI SMT 9.6.2).

fddimibPORTPCWithhold {fddimibPORTEntry 22}
Integer Read-Only

The value of PC_Withhold (corresponds to ANSI {fddiPORT 63}, refer to ANSI SMT 9.4.1).

fddimibPORTLerFlag {fddimibPORTEntry 23}
Integer Read-Only

The condition becomes active when the value of fddiPORTLerEstimate is less than or equal to fddiPORTLerAlarm (corresponds to ANSI {fddiPORT 64}). This will be reported with the Status Report Frames (SRF) (refer to ANSI SMT 7.2.7 and 8.3).

fddimibPORTHardwarePresent {fddimibPORTEntry 24}
Integer Read-Only

This variable indicates the presence of underlying hardware support for this Port object (corresponds to ANSI {fddiPORT 65}). This is always true (1).

fddimibPORTAction {fddimibPORTEntry 25}
Integer Read-Write

A port action to be performed (corresponds to ANSI {fddiPORT 70}). The unit always tries to start the port, if the relevant hardware passes diagnostics, and if the port is configured to be used. Attempts to write to this field with values that are not consistent with the above, will result in “badValue” being returned.

CHAPTER 6
BRIDGE MIB

The unit supports the Bridge MIB {mib-2 17} as defined in *Definitions of Managed Objects for Bridges*, RFC 1286 (Decker, Langille, Rijsinghani, and McCloghrie, editors).

The access types of some of the parameters have been extended from *Read-Only* to *Read-Write*, indicatd by an asterisk (*); other parameters have been changed from *Read-Write* to *Read-Only*, indicated by a number sign (#). The access type of all other parameters is as indicated in the standard MIB.

The MIB tree prefix for reaching the Bridge MIB-II is:

```
1.3.6.1.2.1.
```

dot1dBridge {mib-2 17}

6.1 BASE GROUP

dot1dBase {dot1dBridge 1}

dot1dBaseBridgeAddress {dot1dBase 1}
Octet String Read-Write*

The MAC address used by this bridge when it must be referred to in a unique fashion. It is recommended that this be the numerically smallest MAC address of all ports that belong to this bridge. However it is only required to be unique. When concatenated with dot1dStpPriority a unique BridgeIdentifier is formed which is used in the Spanning Tree Protocol.

The unit's dot1dBaseBridgeAddress is derived from the serial number of the Packet Processing Engine. It is not related to any of the port MAC addresses. Specifying all zeroes for dot1dBaseBridgeAddress will cause the unit to revert to its serial number derivation.

dot1dBaseNumPorts {dot1dBase 2}
Integer Read-Only

The number of ports controlled by this bridging entity.

Identical to ifNumber; however, the unit prevents bridging from being enabled for port 1, which is reserved for out-of-band management.

dot1dBaseType {dot1dBase 3}
Integer Read-Only

Indicates what type of bridging this bridge can perform. If a bridge is actually performing a certain type of bridging this will be indicated by entries in the port table for the given type.

The unit supports srt (4).

dot1dBasePortTable {dot1dBase 4}
Not-Accessible

A table that contains generic information about every port that is associated with this bridge. Transparent, source-route, and srt ports are included.

All of the unit's Ethernet ports are included in this table, regardless of whether any form of bridging is currently enabled for the ports.

dot1dBasePortEntry {dot1dBasePortTable 1}
Not-Accessible

A list of information for each port of the bridge.

dot1dBasePort {dot1dBasePortEntry 1}
Integer Read-Only

The port number of the port for which this entry contains bridge management information.

Identical to dot1dBasePortIfIndex.

dot1dBasePortIfIndex {dot1dBasePortEntry 2}
Integer Read-Only

The value of the instance of the ifIndex object for the interface corresponding to this port.

dot1dBasePortCircuit {dot1dBasePortEntry 3}
Object Identifier Read-Only

For a port which (potentially) has the same value of dot1dBasePortIfIndex as another port on the same bridge, this object contains the name of an object instance unique to this port. For example, in the case where multiple ports correspond one-to-one with multiple X.25 virtual circuits, this value might identify an (e.g., the first) object instance associated with the X.25 virtual circuit corresponding to this port. For a port which has a unique value of dot1dBasePortIfIndex, this object can have the value {0 0}.

Always {0 0}.

dot1dBasePortDelayExceededDiscards {dot1dBasePortEntry 4}
Counter Read-Only

The number of frames discarded by this port due to excessive transit delay through the bridge. It is incremented by both transparent and source route bridges.

dot1dBasePortMtuExceededDiscards {dot1dBasePortEntry 5}
Counter Read-Only

The number of frames discarded by this port due to an excessive size. It is incremented by both transparent and source route bridges.

6.2 SPANNING TREE GROUP

dot1dStp {dot1dBridge 2}

dot1dStpProtocolSpecification {dot1dStp 1}
Integer Read-Only

An indication of what version of the Spanning Tree Protocol is being run. The value decLb100 (2) indicates the DEC LANbridge 100 Spanning Tree protocol. IEEE 802.1d implementations will return ieee8021d (3). If future versions of the IEEE Spanning Tree Protocol are released that are incompatible with the current version a new value will be defined.

Always iee8021d (3).

dot1dStpPriority {dot1dStp 2}
Integer Read-Write

The value of the write-able portion of the Bridge ID, i.e., the first two octets of the (8 octet long) Bridge ID. The other (last) 6 octets of the Bridge ID are given by the value of dot1dBaseBridgeAddress.

Any value from 0 to 65535 may be specified.

dot1dStpTimeSinceTopologyChange {dot1dStp 3}
TimeTicks Read-Only

The time (in hundredths of a second) since the last time a topology change was detected by the bridge entity.

dot1dStpTopChanges {dot1dStp 4}
Counter Read-Only

The total number of topology changes detected by this bridge since the management entity was last reset or initialized.

dot1dStpDesignatedRoot {dot1dStp 5}
Octet String Read-Only

The bridge identifier of the root of the spanning tree as determined by the Spanning Tree Protocol as executed by this node. This value is used as the Root Identifier parameter in all Configuration Bridge PDUs originated by this node.

dot1dStpRootCost {dot1dStp 6}
Integer Read-Only

The cost of the path to the root as seen from this bridge.

dot1dStpRootPort {dot1dStp 7}
Integer Read-Only

The port number of the port which offers the lowest cost path from this bridge to the root bridge.

Port number of this bridge's current spanning tree root port, or 0, if this bridge is the current spanning tree root bridge.

dot1dStpMaxAge {dot1dStp 8}
Integer Read-Only

The maximum age of Spanning Tree Protocol information learned from the network on any port before it is discarded, in units of hundredths of a second. This is the actual value that this bridge is currently using.

dot1dStpHelloTime {dot1dStp 9}
Integer Read-Only

The amount of time between the transmission of Configuration bridge PDUs by this node on any port when it is the root of the spanning tree or trying to become so, in units of hundredths of a second. This is the actual value that this bridge is currently using.

dot1dStpHoldTime {dot1dStp 10}
Integer Read-Only

This time value determines the interval length during which no more than two Configuration bridge PDUs shall be transmitted by this node, in units of hundredths of a second.

dot1dStpForwardDelay {dot1dStp 11}
Integer Read-Only

This time value, measured in units of hundredths of a second, controls how fast a port changes its spanning state when moving towards the Forwarding state. The value determines how long the port stays in a particular state before moving to the next state. For example, how long a port stays in the Listening state when moving from Blocking to Learning. This value is also used, when a topology change has been detected and is underway, to age all dynamic entries in the Forwarding Database. [Note that this value is the one that this bridge is currently using, in contrast to dot1dStpBridgeForwardDelay which is the value that this bridge and all others would start using if/when this bridge were to become the root.]

dot1dStpBridgeMaxAge {dot1dStp 12}
TimeTicks Read-Write

The value that all bridges use for MaxAge when this bridge is acting as the root. Note that 802.1d/D9 specifies that the range for this parameter is related to the value of dot1dStpBridgeHelloTime. The granularity of this timer is specified by 802.1d/D9 to be 1 second. An agent may return a badValue error if a set is attempted to a value which is not a whole number of seconds.

The Max Age Time must be at least 6 seconds, and must adhere to the following equations involving Max Age Time, Forward Delay Time, and Hello Time

2 x (Forward Delay Time - 1) >= Max Age Time

Max Age >= 2 x (Hello Time + 1)

dot1dStpBridgeHelloTime {dot1dStp 13}
TimeTicks Read-Write

The value that all bridges use for HelloTime when this bridge is acting as the root. The granularity of this timer is specified by 802.1d/D9 to be 1 second. An agent may return a badValue error if a set is attempted to a value which is not a whole number of seconds.

The Hello Time must adhere to the equation described in the variable dot1dStpBridgeMaxAge, involving Max Age Time and Hello Time.

dot1dStpBridgeForwardDelay {dot1dStp 14}
TimeTicks Read-Write

The value that all bridges use for ForwardDelay when this bridge is acting as the root. Note that 802.1d/D9 specifies that the range for this parameter is related to the value of dot1dStpBridgeMaxAge. The granularity of this timer is specified by 802.1d/D9 to be 1 second. An agent may return a badValue error if a set is attempted to a value which is not a whole number of seconds.

The Forward Delay Time must be at least 4 seconds, and must adhere to the aforementioned equation involving Max Age Time and Forward Delay Time.

dot1dStpPortTable {dot1dStp 15}
Not-Accessible

A table that contains port-specific information for the Spanning Tree Protocol.

All of the unit's ports are included in this table, regardless of whether the Spanning Tree Protocol is currently enabled for the ports.

dot1dStpPortEntry {dot1dStpPortTable 1}
Not-Accessible

A list of information maintained by every port about the Spanning Tree Protocol state for that port.

dot1dStpPort {dot1dStpPortEntry 1}
Integer Read-Only

The port number of the port for which this entry contains Spanning Tree Protocol management information.

dot1dStpPortPriority {dot1dStpPortEntry 2}
Integer Read-Write

The value of the priority field which is contained in the first (in network byte order) octet of the (2 octet long) Port ID. The other octet of the Port ID is given by the value of dot1dStpPort.

Any value from 0 to 255 may be specified.

dot1dStpPortState {dot1dStpPortEntry 3}
Integer Read-Only

The port's current state as defined by application of the Spanning Tree Protocol. This state controls what action a port takes on reception of a frame. If the bridge has detected a port that is malfunctioning it will place that port into the broken (6) state. For ports which are disabled (see dot1dStpPortEnable), this object will have a value of disabled (1).

Values include:

- disabled (1)
- blocking (2)
- listening (3)
- learning (4)
- forwarding (5)
- broken (6)

dot1dStpPortEnable {dot1dStpPortEntry 4}
Integer Read-Write

The enabled/disabled status of the port.

The unit does not change the parameter's value when processing a SetRequest - refer to the sprotoBridge parameter in Cabletron's enterprise-specific MIB for the SetRequest capabilities.

dot1dStpPortPathCost {dot1dStpPortEntry 5}
Integer Read-Write

The contribution of this port to the path cost of paths towards the spanning tree root which include this port.

Any value from 0 to 65535 may be specified; specifying a value of 0 will cause the unit to automatically compute the proper default value. [1]

[1.] The proper default is the inverse of the media baud rate (e.g., 10 for FDDI, 100 for Ethernet). The effective value is always returned in the GetResponse. This is useful primarily for 802.5 Token Ring, where the recommended cost weighting varies for 4 Mbps rings.

dot1dStpPortDesignatedRoot {dot1dStpPortEntry 6}
Octet String Read-Only

The unique Bridge Identifier of the Bridge recorded as the Root in the Configuration BPDUs transmitted by the Designated Bridge for the segment to which the port is attached.

dot1dStpPortDesignatedCost {dot1dStpPortEntry 7}
Integer Read-Only

The path cost of the Designated Port of the segment connected to this port. This value is compared to the Root Path Cost field in received bridge PDUs.

dot1dStpPortDesignatedBridge {dot1dStpPortEntry 8}
Octet String Read-Only

The Bridge Identifier of the bridge which this port considers to be the Designated Bridge for this port's segment.

dot1dStpPortDesignatedPort {dot1dStpPortEntry 9}
Octet String Read-Only

The Port Identifier of the port on the Designated Bridge for this port's segment.

dot1dStpPortForwardTransitions {dot1dStpPortEntry 10}
Counter Read-Only

The number of times this port has transitioned from the Learning state to the Forwarding state.

6.3 SOURCE ROUTE GROUP

dot1dSr {dot1dBridge 3}

dot1dSrPortTable {dot1dSr 1}
Not-Accessible

A table that contains information about every port that is associated with this source route bridge.

Only those ports that are enabled for source routing (or source routing transparent) bridging appear in this table. Refer to Cabletron's enterprise-specific MIB sprotoBridge parameter for the means to enable/disable source routing on a per-port basis.

dot1dSrPortEntry {dot1dSrPortTable 1}
Not-Accessible

A list of information for each port of a source route bridge.

dot1dSrPort {dot1dSrPortEntry 1}
Integer Read-Only

The port number of the port for which this entry contains Source Route management information.

dot1dSrPortHopCount {dot1dSrPortEntry 2}
Integer Read-Write

The maximum number of routing descriptors allowed in an All Paths or Spanning Tree Explorer frames.

Any value from 0 to 8 may be specified; specifying 0 will cause the unit to return its default value_.[2]

dot1dSrPortLocalSegment {dot1dSrPortEntry 3}
Integer Read-Write

The segment number that uniquely identifies the segment to which this port is connected. Current source routing protocols limit this value to the range 0 through 4095. A value of 65535 signifies that no segment number is assigned to this port.

dot1dSrPortBridgeNum {dot1dSrPortEntry 4}
Integer Read-Write

A bridge number uniquely identifies a bridge when more than one bridge is used to span the same two segments. Current source routing protocols limit this value to the range 0 through 15. A value of 65535 signifies that no bridge number is assigned to this bridge.

---

2. The unit's default value is 8; however, in the future, the default value may be changed to 14 in order to conform to the SRT standard.

dot1dSrPortTargetSegment {dot1dSrPortEntry 5}
Integer Read-Write

The segment number that corresponds to the target segment this port is considered to be connected to by the bridge. Current source routing protocols limit this value to the range 0 through 4095. A value of 65535 signifies that no target segment is assigned to this port.

dot1dSrPortLargestFrame {dot1dSrPortEntry 6}
Integer Read-Write

The maximum size of the INFO field (LLC and above) that this port can send/receive. It does not include any MAC level (framing) octets. The value of this object is used by this bridge to determine whether a modification of the LargestFrame field of the Routing Control field of the Routing Information Field is necessary. Valid values as defined by the 802.5 source routing bridging specification are 516, 1500, 2052, 4472, 8144, 11407, 17800, and 65535 octets. Behavior of the port when an illegal value is written is implementation specific. It is recommended that a reasonable legal value be chosen.

dot1dSrPortSTESpanMode {dot1dSrPortEntry 7}
Integer Read-Write

Determines how this port behaves when presented with a Spanning Tree Explorer frame. The value disabled (2) indicates that the port will not accept or send Spanning Tree Explorer packets; any STE packets received will be silently discarded. The value forced (3) indicates the port will always accept and propagate Spanning Tree Explorer frames. This allows a manually configured Spanning Tree for this class of packet to be configured. Note that unlike transparent bridging this is not catastrophic to the network if there are loops. The value auto-span (1) can only be returned by a bridge that both implements the Spanning Tree Protocol and has use of the protocol enabled on this port. The behavior of the port for Spanning Tree Explorer frames is determined by the state of dot1dStpPortState. If the port is in the forwarding state, the frame will be accepted or propagated. Otherwise it will be silently discarded.

dot1dSrPortSpecInFrames {dot1dSrPortEntry 8}
Counter Read-Only

The number of specifically routed frames that have been received from this port's segment.

dot1dSrPortSpecOutFrames {dot1dSrPortEntry 9}
Counter Read-Only

The number of specifically routed frames that this port has transmitted on its segment.

dot1dSrPortApeInFrames {dot1dSrPortEntry 10}
Counter Read-Only

The number of all paths explorer frames that have been received by this port from its segment.

dot1dSrPortApeOutFrames {dot1dSrPortEntry 11}
Counter Read-Only

The number of all paths explorer frames that have been transmitted by this port on its segment.

dot1dSrPortSteInFrames {dot1dSrPortEntry 12}
Counter Read-Only

The number of spanning tree explorer frames that have been received by this port from its segment.

dot1dSrPortSteOutFrames {dot1dSrPortEntry 13}
Counter Read-Only

The number of spanning tree explorer frames that have been transmitted by this port on its segment.

dot1dSrPortSegmentMismatchDiscards {dot1dSrPortEntry 14}
Counter Read-Only

The number of explorer frames that have been discarded by this port because the routing descriptor field contained an invalid adjacent segment value.

dot1dSrPortDuplicateSegmentDiscards {dot1dSrPortEntry 15}
Counter Read-Only

The number of frames that have been discarded by this port because the routing descriptor field contained a duplicate segment identifier.

dot1dSrPortHopCountExceededDiscards {dot1dSrPortEntry 16}
Counter Read-Only

The number of explorer frames that have been discarded by this port because the Routing Information Field has exceeded the maximum route descriptor length.

6.4 TRANSPARENT GROUP

dot1dTp {dot1dBridge 4}

dot1dTpLearnedEntryDiscards {dot1dTp 1}
Counter Read-Only

The total number of Forwarding Database entries, which have been or would have been learned, but have been discarded due to a lack of space to store them in the Forwarding Database. If this counter is increasing, it indicates that the Forwarding Database is regularly becoming full (a condition which has unpleasant performance effects on the subnetwork). If this counter has a significant value but is not presently increasing, it indicates that the problem has been occurring but is not persistent.

dot1dTpAgingTime {dot1dTp 2}
Integer Read-Write

The timeout period in seconds for aging out dynamically learned forwarding information.

dot1dTpFdbTable {dot1dTp 3}
Not-Accessible

A table that contains information about unicast entries for which the bridge has forwarding and/or filtering information. This information is used by the transparent bridging function in determining how to propagate a received frame.

A superset of this table's information can be found by using Cabletron's enterprise-specific Addresses Configuration Group.

dot1dTpFdbEntry {dot1dTpFdbTable 1}
Not-Accessible

Information about a specific unicast MAC address for which the bridge has some forwarding and/or filtering information.

dot1dTpFdbAddress {dot1dTpFdbEntry 1}
Octet String Read-Only

A unicast MAC address for which the bridge has forwarding and/or filtering information.

dot1dTpFdbPort {dot1dTpFdbEntry 2}
Integer Read-Only

Either the value 0, or the port number of the port on which a frame having a source address equal to the value of the corresponding instance of dot1dTpFdbAddress has been seen. A value of 0 indicates that the port number has not been learned but that the bridge does have some forwarding/filtering information about this address (e.g., in the dot1dStaticTable). Implementors are encouraged to assign the port value to this object whenever it is learned even for addresses for which the corresponding value of dot1dTpFdbStatus is not learned (3).

dot1dTpFdbStatus {dot1dTpFdbEntry 3}
Integer Read-Only

The status of this entry. The meanings of the values are other (1) none of the following. This would include the case where some other MIB object (not the corresponding instance of dot1dTpFdbPort, nor an entry in the dot1dStaticTable) is being used to determine if and how frames addressed to the value of the corresponding instance of dot1dTpFdbAddress are being forwarded. invalid (2) this entry is not longer valid (e.g., it was learned but has since aged-out), but has not yet been flushed from the table. learned (3) the value of the corresponding instance of dot1dTpFdbPort was learned, and is being used. self (4) the value of the corresponding instance of dot1dTpFdbAddress represents one of the bridge's addresses. The corresponding instance of dot1dTpFdbPort indicates which of the bridge's ports has this address. mgmt (5) the value of the corresponding instance of dot1dTpFdbAddress is also the value of an existing instance of dot1dStaticAddress.

dot1dTpPortTable {dot1dTp 4}
Not-Accessible

A table that contains information about every port that is associated with this transparent bridge.

Only those ports that are enabled for transparent (or source routing transparent) bridging appear in this table. Refer to Cabletron's enterprise-specific MIB sprotoBridge parameter for the means to enable/disable transparent bridging on a per-port basis.

dot1dTpPortEntry {dot1dTpPortTable 1}
Not-Accessible

A list of information for each port of a transparent bridge.

dot1dTpPort {dot1dTpPortEntry 1}
Integer Read-Only

The port number of the port for which this entry contains transparent bridging management information.

dot1dTpPortMaxInfo {dot1dTpPortEntry 2}
Integer Read-Only

The maximum size of the INFO (non-MAC) field that this port will receive or transmit.

The type/length field (2 octets) in Ethernet/802.3 packets are considered to be part of the MAC header.

dot1dTpPortInFrames {dot1dTpPortEntry 3}
Counter Read-Only

The number of frames that have been received by this port from its segment. Note that a frame received on the interface corresponding to this port is only counted by this object if and only if it is for a protocol being processed by the local bridging function.

dot1dTpPortOutFrames {dot1dTpPortEntry 4}
Counter Read-Only

The number of frames that have been transmitted by this port to its segment. Note that a frame transmitted on the interface corresponding to this port is only counted by this object if and only if it is for a protocol being processed by the local bridging function.

dot1dTpPortInDiscards {dot1dTpPortEntry 5}
Counter Read-Only

Count of valid frames received which were discarded (i.e., filtered) by the Forwarding Process.

Local traffic is usually discarded before it reaches the Forwarding Process (see sifFilterLocal).

6.5 STATIC DATABASE GROUP

dot1dStatic {dot1dBridge 5}

dot1dStaticTable {dot1dStatic 1}
Not-Accessible

A table containing filtering information configured into the bridge by (local or network) management specifying the set of ports to which frames received from specific ports and containing specific destination addresses are allowed to be forwarded. The value of zero in this table as the port number from which frames with a specific destination address are received, is used to specify all ports for which there is no specific entry in this table for that particular destination address. Entries are valid for unicast and for group/broadcast addresses.

The unit does not store the value zero for port number. A value of zero is not valid for GetRequests. For SetRequests a value of zero will cause entries to be created for all ports that have no entries defined.

A superset of this table's information can be found by using Cabletron's enterprise-specific Addresses Configuration Group.

dot1dStaticEntry {dot1dStaticTable 1}
Not-Accessible

Filtering information configured into the bridge by (local or network) management specifying the set of ports to which frames received from a specific port and containing a specific destination address are allowed to be forwarded.

The index for an entry is dot1dStaticAddress and dot1dStaticReceivePort.

dot1dStaticAddress {dot1dStaticEntry 1}
Octet String Read-Write

The destination MAC address in a frame to which this entry's filtering information applies. This object can take the value of a unicast address, a group address or the broadcast address.

dot1dStaticReceivePort {dot1dStaticEntry 2}
Integer Read-Write

Either the value 0, or the port number of the port from which a frame must be received in order for this entry's filtering information to apply. A value of zero indicates that this entry applies on all ports of the bridge for which there is no other applicable entry.

The unit does not store the value 0. Instead, a SetRequest with a value 0 will cause the unit to automatically create entries for all ports that have no other applicable entries.

dot1dStaticAllowedToGoTo {dot1dStaticEntry 3}
Octet String Read-Write

The set of ports to which frames received from a specific port and destined for a specific MAC address, are allowed to be forwarded. Each octet within the value of this object specifies a set of eight ports, with the first octet specifying ports 1 through 8, the second octet specifying ports 9 through 16, etc. Within each octet, the most significant bit represents the lowest numbered port, and the least significant bit represents the highest numbered port. Thus, each port of the bridge is represented by a single bit within the value of this object. If that bit has a value of 1 then that port is included in the set of ports; the port is not included if its bit has a value of 0. [Note that the setting of the bit corresponding to the port from which a frame is received is irrelevant.]

On the unit, each static address entry exists internally as part of a full matrix of entry ports and “allowed-to-go” exit ports. The value of dot1dStaticAllowedToGoTo corresponds to one row of the matrix. Creation of the first entry causes the entire matrix to be created. Rows are created for all other entries with dot1dStaticAllowedToGoTo values of zero. Note that the creation of a static entry for an entry port causes the MAC address to be treated according to the matrix and precludes any dynamic learning of that MAC address.

dot1dStaticStatus {dot1dStaticEntry 4}
Integer Read-Write

This object indicates the status of this entry. other (1) - this entry is currently in use but the conditions under which it will remain so are different from each of the following values. invalid (2) - writing this value to the object removes the corresponding entry. permanent (3) - this entry is currently in use and will remain so after the next reset of the bridge. deleteOnReset (4) - this entry is currently in use and will remain so until the next reset of the bridge. deleteOnTimeout (5) - this entry is currently in use and will remain so until it is aged out.

Static entries are always permanent (3). SetRequests may only specify invalid (2), in which case, the entry will be immediately deleted. Deleting an entry causes all entries with the dot1dStaticAddress to be deleted, regardless of the dot1dStaticReceivePort.

CHAPTER 7
REPEATER MIB

The unit supports the Ethernet Repeater MIB as defined in *Definitions of Managed Objects for IEEE 802.3 Repeater Devices*. RFC 1368 (McMaster and McCloghrie, editors).

The unit has one major special feature. Packets are not repeated across Groups. Instead, Groups are treated as separate repeaters, which may be logically connected by the bridging/routing functions performed by the unit.

The MIB tree for reaching the Ethernet Repeater group is:

```
1.3.6.1.2.1.
```

snmpDot3RptrMgt {mib-2 22}

7.1 BASIC PACKAGE

rptrBasicPackage {snmpDot3RptrMgt 1}

rptrRptrInfo {rptrBasicPackage 1}

rptrGroupCapacity {rptrRptrInfo 1}
Integer Read-Only

The rptrGroupCapacity is the number of groups that can be contained within the repeater. Within each managed repeater, the groups are uniquely numbered in the range from 1 to rptrGroupCapacity. Some groups may not be present in the repeater, in which case the actual number of groups present will be less than rptrGroupCapacity. Note in practice, this will generally be the number of field-replaceable units (i.e., modules, cards, or boards) that can fit in the physical repeater enclosure, and the group numbers will correspond to numbers marked on the physical enclosure.

Since the unit has a maximum of five I/O modules, this number is always five.

rptrOperStatus {rptrRptrInfo 2}
Integer Read-Only

The rptrOperStatus object indicates the operational state of the repeater. The rptrHealthText object may be consulted for more specific information about the state of the repeater's health. In the case of multiple kinds of failures (e.g., repeater failure and port failure), the value of this attribute shall reflect the highest priority failure in the following order rptrFailure (3), groupFailure (4), portFailure (5), generalFailure (6).

The unit returns either ok (1) or portFailure (5).

rptrHealthText {rptrRptrInfo 3}
Octet String Read-Only

The health text object is a text string that provides information relevant to the operational state of the repeater. Agents may use this string to provide detailed information on current failures, including how they were detected, and/or instruction for problem resolution.

The unit always returns a zero-length string.

rptrReset {rptrRptrInfo 4}
Integer Read-Write

Setting this object to reset (2) causes a transition to the START state. Setting this object to noReset (1) has no effect. The agent will always return the value noReset (1) when this object is read. This action does not reset the management counters defined in this document nor does it affect the portAdminStatus parameters. Included in this action is the execution of a disruptive Self-Test with the following characteristics a) The nature of the tests is not specified. b) The test resets the repeater but without affecting management information about the repeater. c) The test does not inject packets onto any segment. d) Packets received during the test may or may not be transferred. e) The test does not interfere with management functions. As a result of this action a rptrResetEvent trap should be sent.

The unit provides no mechanism for implementing any reset of the concentrator other than power-up. Setting this object to reset (2), will result in a genErr response.

rptNonDisruptTest {rptrRptrInfo 5}
Integer Read-Write

Setting this object to selfTest (2) causes the repeater to perform a agent-specific, non-disruptive self-test that has the following characteristics a) The nature of the tests is not specified. b) The test does not change the state of the repeater or management information about the repeater. c) The test does not inject packets onto any segment. d) The test does not prevent the relay of any packets. e) The test does not interfere with management functions. After performing this test the agent will update the repeater health information and send a rptrHealth trap. Setting this object to noSelfTest (1) has no effect. The agent will always return the value noSelfTest (1) when this object is read.

This test does nothing.

rptrTotalPartitionedPorts {rptrRptrInfo 6}
Gauge Read-Only

Returns the total number of ports in the repeater whose current state meets all three of the following criteria: rptrPortOperStatus does not have the value notPresent (3), rptrPortAdminStatus is enabled (1), and rptrPortAutoPartitionState is autoPartitioned (2).

7.2 GROUP INFO

rptrGroupInfo {rptrBasicPackage 2}

rptrGroupTable {rptrGroupInfo 1}
Not-Accessible

Table of descriptive and status information about the groups of ports.

rptrGroupEntry {rptrGroupTable 1}
Not-Accessible

An entry in the table, containing information about a single group of ports.

rptrGroupIndex {rptrGroupEntry 1}
Integer Read-Only

This object identifies the group within the repeater for which this entry contains information. This value is never greater than rptrGroupCapacity.

rptrGroupDescr {rptrGroupEntry 2}
Octet String Read-Only

A textual description of the group. This value should include the full name and version identification of the group's hardware type and indicate how the group is differentiated from other groups in the repeater. Plug in Module, "Rev A" or "John Doe 10BASE-T 4-port SIMM socket Version 2.1" are examples of valid group descriptions. It is mandatory that this only contain printable ASCII characters.

This is "CEIOM (n) part-no", where "n" is the module id number (i.e., a value from 2 to 6; see hwIndex), and "part-no" is the 24 character manufacturing part/rev/serial number.

rptrGroupObjectID {rptrGroupEntry 3}
Object Identifier Read-Only

The vendor's authoritative identification of the group. This value is allocated within the Cabletron enterprises subtree (1.3.6.1.4.1) and provides a straight-forward and unambiguous means for determining what kind of group is being managed. For example, this object could take the value 1.3.6.1.4.1.4242.1.2.14 if vendor "Doe, Inc." was assigned the subtree 1.3.6.1.4.1.4242, and had assigned the identifier 1.3.6.1.4.1.4242.1.2.14 to its "Mary Doe 6-Port FOIRL Plug-in Module."

This value is {sysID a b c}, where "a" is the manufacturing class number of the CEIOM, "b" is the CEIOM's sequence number within its class, and "c" is the CEIOM's revision level.

rptrGroupOperStatus {rptrGroupEntry 4}
Integer Read-Only

An object that indicates the operational status of the group. A status of notPresent (4) indicates the group is temporarily or permanently physically and/or logically not a part of the repeater. It is an implementation-specific matter as to whether the agent effectively removes notPresent entries from the table. A status of operational (2) indicates that the group is functioning, and a status of malfunctioning (3) indicates that the group is malfunctioning in some way.

The unit automatically removes notPresent entries, and always returns either operational (2) or other (1).

rptrGroupLastOperStatusChange {rptrGroupEntry 5}
TimeTicks Read-Only

An object that contains the value of sysUpTime at the time that the value of the rptrGroupOperStatus object for this group last changed. A value of zero indicates that the group's oper status has not changed since the agent last restarted.

rptrGroupPortCapacity {rptrGroupEntry 6}
Integer Read-Only

The rptrGroupPortCapacity is the number of ports that can be contained within the group. Valid range is 1-1024. Within each group, the ports are uniquely numbered in the range from 1 to rptrGroupPortCapacity. Note In practice, this will generally be the number of ports on a module, card, or board, and the port numbers will correspond to numbers marked on the physical embodiment.

7.3 PORT INFO

rptrPortInfo {rptrBasicPackage 3}

rptrPortTable {rptrPortInfo 1]
Not-Accessible

Table of descriptive and status information about the ports.

rptrPortEntry {rptrPortTable 1}
Not-Accessible

An entry in the table, containing information about a single port.

rptrPortGroupIndex {rptrPortEntry 1}
Integer Read-Write

This object identifies the group containing the port for which this entry contains information.

rptrPortIndex {rptrPortEntry 2}
Integer Read-Only

This object identifies the port within the group for which this entry contains information. this value can never be greater than rptrGroupPortCapacity for the associated group.

rptrPortAdminStatus {rptrPortEntry 3}
Integer Read-Write

Setting this object to disabled (2) disables the port. A disabled port neither transmits nor receives. Once disabled, a port must be explicitly enabled to restore operation. A port which is disabled when power is lost or when a reset is exerted shall remain disabled when normal operation resumes. The admin status takes precedence over auto-partition and functionally operates between the auto-partition mechanism and the AUI/PMA. Setting this object to enabled (1) enables the port and exerts a BEGIN on the port's auto-partition state machine. (In effect, when a port is disabled, the value of rptrPortAutoPartitionState for that port is frozen until the port is next enabled. When the port becomes enabled, the rptrPortAutoPartitionState becomes notAutoPartitioned (1), regardless of its pre-disabling state.

rptrPortAutoPartitionState {rptrPortEntry 4}
Integer Read-Only

The autoPartitionState flag indicates whether the port is currently partitioned by the repeater's auto-partition protection. The conditions that cause port partitioning are specified in partition state machine in section 9 (IEEE 802.3 std.). They are not differentiated here.

rptrPortOperStatus {rptrPortEntry 5}
Integer Read-Only

This object indicates the port's operational status. The notPresent (3) status indicates the port is physically removed (note this may or may not be possible depending on the type of port). The operations (1) status indicates that the port is enabled (see rptrPortAdminStatus) and working, even though it might be auto-partitioned (see rptrPortAutoPartitionState). If this object has the value operational (1) and rptrPortAdminStatus is set to disabled (2), it is expected that this object's value will change to notOperational (2) soon after.

The unit returns either operational (1) or notOperational (2).

7.4 MONITOR PACKAGE

rptrMonitorPackage {snmpDot3RptrMgt 2}

7.5 MONITOR RPTR INFO

rptrMonitorRptrInfo {rptrMonitorPackage 1}

rptrMonitorTransmitCollisions {rptrMonitorRptrInfo 1}
Counter Read-Only

This counter is incremented every time the repeater state machine enters the TRANSMIT COLLISION state from any state other than ONE PORT LEFT. (Ref Fig. 9-2, IEEE 802.3 Std). The approximate minimum time for rollover of this counter is 16 hours.

7.6 MONITOR GROUP INFO

rptrMonitorGroupInfo {rptrMonitorPackage 2}

rptrMonitorGroupTable {rptrMonitorGroupInfo 1}
Not-Accessible

Table of performance and error statistics for the groups.

rptrMonitorGroupEntry {rptrMonitorGroupTable 1}
Not-Accessible

An entry in the table containing total performance and error statistics for a single group. Regular retrieval of the information in this table provides a means of tracking the performance and health of the networked devices attached to this group's ports. The counters in this table are redundant in the sense that they are the summations of information already available through other objects. However, these sums provide a considerable optimization of network management traffic over the otherwise necessary retrieval of the individual counters included in each sum.

rptrMonitorGroupIndex {rptrMonitorGroupEntry 1}
Integer Read-Only

This object identifies the group within the repeater for which this entry contains information.

rptrMonitorGroupTotalFrames {rptMonitorGroupEntry 2}
Integer Read-Only

The total number of frames of valid frame length that have been received on the ports in this group. This counter is the summation of the values of the rptrMonitorPortReadableFrames counters for all of the ports in the groups. This statistic provides one of the parameters necessary for obtaining the packet error rate. The approximate minimum time for rollover of this counter is 80 hours.

rptrMonitorGroupTotalOctets {rptrMonitorGroupEntry 3}
Counter Read-Only

The total number of octets contained in the valid frames that have been received on the ports in this group. This statistic provides an indicator of the total data transferred. The approximate minimum time for rollover of this counter is 58 minutes.

rptrMonitorGroupTotalErrors {rptrMonitorGroupEntry 4}
Counter Read-Only

The total number of errors which have occurred on all of the ports in this group. This counter is the summation of the values of the rptrMonitorPortTotalErrors counters for all of the ports in the group.

7.7 MONITOR PORT INFO

rptrMonitorPortInfo {rptrMonitorPackage 3}

rptrMonitorPortTable {rptrMonitorPortInfo 1}
Not-Accessible

Table of performance and error statistics for the ports.

rptrMonitorPortEntry {rptrMonitorPortTable 1}
Not-Accessible

An entry in the table, containing performance and error statistics for a single port.

rptrMonitorPortGroupIndex {rptrMonitorPortEntry 1}
Integer Read-Only

This object identifies the group containing the port for which this entry contains information.

rptrMonitorPortIndex {rptrMonitorPortEntry 2}
Integer Read-Only

This object identifies the port within the group for which this entry contains information.

rptrMonitorPortReadableFrames {rptrMonitorPortEntry 3}
Counter Read-Only

This object is the number of frames of valid frame length that have been received on this port. this counter is incremented by one for each frame received on this port whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (Ref IEEE 802.3 Std, 4.4.2.1) and for which the FCSError and CollisionEvent signals are not asserted. This statistic provide one of the parameters necessary for obtaining the packet error rate. The approximate minimum time for rollover of this counter is 80 hours.

rptrMonitorPortReadableOctets {rptrMonitorPortEntry 4}
Counter Read-Only

This object is the number of octets contained in valid frames length that have been received on this port. This counter is incremented by OctetCount for each frame received on this port which has been determined to be a readable frame. This statistic provides an indicator of the total data transferred. The approximate minimum time for rollover of this counter is 58 minutes.

rptrMonitorPortFCSErrors {rptrMonitorPortEntry 5}
Counter Read-Only

This counter is incremented by one for each frame received on this port with the FCSError signal asserted and the FramingError and CollisionEvent signals deasserted and whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (Ref 4.4.2.1, IEEE 802.3 Std). The approximate minimum time for rollover of this counter is 80 hours.

rptrMonitorPortAlignmentErrors {rptrMonitorPortEntry 6}
Counter Read-Only

This counter is incremented by one for each frame received on this port with the FCSError and FramingError signals asserted and CollisionEvent signal deasserted and whose OctetCount is greater than or equal to minFrameSize and less than or equal to maxFrameSize (Ref IEEE 802.3 Std, 4.4.2.1). If rptrMonitorPortAlignmentErrors is incremented then the rptrMonitorPortFCSErrors Counter shall not be incremented for the same frame. The approximate minimum time for rollover of this counter is 80 hours.

rptrMonitorPortFrameTooLongs {rptrMonitorPortEntry 7}
Counter Read-Only

This counter is incremented by one for each frame received on this port whose OctetCount is greater than maxFrameSize (Ref 4.4.2.1, IEEE 802.3 Std). If rptrMonitorPortFrameTooLongs is incremented then neither the rptrMonitorPortFCSErrors Counter nor the rptrMonitorPortAlignmentErrors Counter shall be incremented for the same frame. The approximate minimum time for rollover of this counter is 61 days.

rptrMonitorPortShortEvents {rptrMonitorPortEntry 8}
Counter Read-Only

This counter is incremented by one for each CarrierEvent on this port with ActivityDuration less than ShortEventMaxTime. ShortEventMaxTime is greater than 74 bit times and less than 82 bit times. ShortEventMaxTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state machine. Note shortEvents may indicate externally generated noise bits which will cause the repeater to transmit Runts to its other ports, or propagate a collision (which may be late) back to the transmitting DTE and damaged frames to the rest of the network. Implementors may wish to consider selecting the ShortEventMaxTime towards the lower end of the allowed tolerance range to accommodate bit losses suffered through physical channel devices not budgeted for within this standard. The approximate minimum time for rollover of this counter is 16 hours.

rptrMonitorPortRunts {rptrMonitorPortEntry 9}
Counter Read-Only

This counter is incremented by one for each CarrierEvent on this port that meets one of the following two conditions. Only one test need be made. a) The ActivityDuration is greater than ShortEventMaxTime and less than ValidPacketMinTime and the CollisionEvent signal is deasserted. b) The OctetCount is less than 64, the ActivityDuration is greater than ShortEventMaxTime and the CollisionEvent signal is deasserted. ValidPacketMinTime is greater than or equal to 552 bit times and less than 565 bit times. An event whose length is greater than 74 bit times but less than 82 bit times shall increment either the shortEvents counter or the runts counter but not both. A CarrierEvent greater than or equal to 552 bit times but less than 565 bit times may or may not be counted as a runt. ValidPacketMinTime has tolerances included to provide for circuit losses between a conformance test point at the AUI and the measurement point within the state machine. Runts usually indicate collision fragments, a normal network event. In certain situations associated with large diameter networks a percentage of runts may exceed ValidPacketMinTime. The approximate minimum time for rollover of this counter is 16 hours.

rptrMonitorPortCollisions {rptrMonitorPortEntry 10}
Counter Read-Only

This counter is incremented by one for any CarrierEvent on any port for which the CollisionEvent Signal on this port is asserted. The approximate minimum time for rollover of this counter is 16 hours.

rptrMonitorPortLateEvents {rptrMonitorPortEntry 11}
Counter Read-Only

This counter is incremented by one for each CarrierEvent on this port in which the Collin (X) variable transitions to the value SQE (Ref 9.6.6.2, IEEE 802.3 Std) while the ActivityDuration is greater than the LateEventThreshold. Such a CarrierEvent is counted twice, as both a collision and as a lateEvent. The LateEventTheshold is greater than 480 bit times and less than 565 bit times. LateEventThreshold has tolerances included to permit an implementation to build a single threshold to serve as both the LateEventThreshold and ValidPacketMinTime threshold. The approximate minimum time for rollover of this counter is 81 hours.

rptrMonitorPortVeryLongEvents {rptrMonitorPortEntry 12}
Counter Read-Only

This counter is incremented by one for each CarrierEvent on this port whose ActivityDuration is greater than the MAU Jabber Lockup Protection timer TW3 (Ref 9.6.1 & 9.6.5, IEEE 802.3 Std). Other counters may be incremented as appropriate.

rptrMonitorPortDataRateMismatches {rptrMonitorPortEntry 13}
Counter Read-Only

This counter is incremented by one for each frame received on this port that meets all of the following conditions: a) The CollisionEvent signal is not asserted. b) The ActivityDuration is greater than ValidPacketMinTime. c) The frequency (data rate) is detectably mismatched from the local transmit frequency. The exact degree of mismatch is vendor specific and is to be defined by the vendor for conformance testing. When this event occurs, other counters whose increment conditions were satisfied may or may not also be incremented, at the implementor's discretion. Whether or not the repeater was able to maintain data integrity is beyond the scope of this standard.

This is polled from the RIC event-record-register ELBER (elasticity buffer error) bit, ten times per second. As such, it is subject to under-counting.

rptrMonitorPortAutoPartitions {rptrMonitorPortEntry 14}
Counter Read-Only

This counter is incremented by one for each time the repeater has automatically partitioned this port. The conditions that cause port partitioning are specified in the partition state machine in Section 9 (IEEE 802.3 Std). They are not differentiated here.

This is polled from the RIC event-record-register PART bit, ten times per second. As such, it is subject to under-counting.

rptrMonitorPortTotalErrors {rptrMonitorPortEntry 15}
Counter Read-Only

The total number of errors which have occurred on this port. This counter is the summation of the values of other error counters (for the same port), namely rptrMonitorPortFCSErrors, rptrMonitorPortAlignmentErrors, rptrMonitorPortFrameTooLongs, rptrMonitorPortShortEvents, rptrMonitorPortLateEvents, rptrMonitorPortVeryLongEvents, and rptrMonitorPortDataRateMismatches. This counter is redundant in the sense that it is the summation of information already available through other objects. However, it is included specifically because the regular retrieval of this object as a means of tracking the health of a port provides a considerable optimization of network management traffic over the otherwise necessary retrieval of the summed counters.

7.8 ADDRESS TRACKING PACKAGE

{snmpDot3RptrMgt 3}

7.9 ADDRESS TRACKING RPTR INFO

{rptrAddrTrackPackage 1}

This subtree is currently undefined.

7.10 ADDRESS TRACKING GROUP INFO

{rptrAddrTrackPackage 2}

This subtree is currently undefined.

7.11 ADDRESS TRACKING PORT INFO

rptrAddrTrackPortInfo {rptrAddrTrackPackage 2}

rptrAddrTrackTable {rptrAddrTrackPortInfo 1}
Not-Accessible

Table of address mapping information about the ports.

rptrAddrTrackEntry {rptrAddrTrackTable 1}
Not-Accessible

An entry in the table, containing address mapping information about a single port.

rptrAddrTrackGroupIndex {rptrAddrTrackEntry 1}
Integer Read-Only

This object identifies the group containing the port for which this entry contains information.

rptrAddrTrackPortIndex {rptrAddrTrackEntry 2}
Integer Read-Only

This object identifies the port within the group for which this entry contains information.

rptrAddrTrackLastSourceAddress {rptrAddrTrackEntry 3}
Octet String Read-Only

This object is the SourceAddress of the last readable frame (i.e., counted by rptrMonitorPortReadableFrames) received by this port.

rptrAddrTrackSourceAddrChanges {rptrAddrTrackEntry 4}
Counter Read-Only

This counter is incremented by one for each time that the rptrAddrTrackLastSourceAddress attribute for this port has changed. This may indicate whether a link is connected to a single DTE or another multi-user segment. The approximate minimum time for rollover of this counter is 81 hours.

CHAPTER 8
PPP MIB

The unit supports RFC 1471 for PPP link control and RFC 1473 for PPP IP, and RFC 1474 for PPP Bridge support.

The MIB tree for reaching the PPP group is:

```
1.3.6.1.2.1.10.
```

ppp {transmission 23}

8.1 PPP LINK CONTROL TABLE

pppLCP {ppp 1}

pppLink {pppLCP 1}

pppLinkStatusTable {pppLink 1}
Not-Accessible

A table containing PPP-Link specific variables for this PPP implementation.

pppLinkStatusEntry {pppLinkStatusTable 1}
Not-Accessible

Management information about a particular PPP Link.

pppLinkStatusPhysicalIndex {pppLinkStatusEntry 1}
Integer Read-only

The value of ifIndex that identifies the lower-level interface over which this PPP Link is operating. This interface would usually be an HDLC or RS-232 type of interface. If there is no lower layer interface element, or there is no ifEntry for the element, or the element can not be identified, then the value of this object is 0. For example, suppose that PPP is operating over a serial port. This would use two entries in the ifTable. The PPP could be running over “interface” number 123 and the serial port could be running over “interface” number 987. Therefore, ifSpecific.123 would contain the OBJECT IDENTIFIER ppp, pppLinkStatusPhysicalIndex.123 would contain 987, and ifSpecific.987 would contain the OBJECT IDENTIFIER for the serial-port’s media-specific MIB.

pppLinkStatusBadAddresses {pppLinkStatusEntry 2}
Counter Read-only

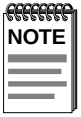
The number of packets received with an incorrect address field. This counter is a component of the ifInErrors variable that is associated with the interface that represents this PPP Link.

pppLinkStatusBadControls {pppLinkStatusEntry 3}
Counter Read-only

The number of packets received on this link with an incorrect control field. This counter is a component of the ifInErrors variable that is associated with the interface that represents this PPP Link.

pppLinkStatusPacketTooLongs {pppLinkStatusEntry 4}
Counter Read-only

The number of received packets that have been discarded because their length exceeded the MRU. This counter is a component of the ifInErrors variable that is associated with the interface that represents this PPP Link.



Packets which are longer than the MRU but which are successfully received and processed are NOT included in this count.

pppLinkStatusBadFCSs {pppLinkStatusEntry 5}
Counter Read-only

The number of received packets that have been discarded due to having an incorrect FCS. This counter is a component of the ifInErrors variable that is associated with the interface that represents this PPP Link.

pppLinkStatusLocalMRU {pppLinkStatusEntry 6}
Integer Read-only

The current value of the MRU for the local PPP Entity. This value is the MRU that the remote entity is using when sending packets to the local PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

Set to 8192 for the maximum HIOM frame size.

pppLinkStatusRemoteMRU {pppLinkStatusEntry 7}
Integer Read-only

The current value of the MRU for the remote PPP Entity. This value is the MRU that the local entity is using when sending packets to the remote PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

pppLinkStatusLocalToPeerACCMap {pppLinkStatusEntry 8}
Octet String Read-only

The current value of the ACC Map used for sending packets from the local PPP entity to the remote PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

Value of 0 indicates that ACC Map is not supported.

pppLinkStatusPeerToLocalACCMap {pppLinkStatusEntry 9}
Octet String Read-only

The ACC Map used by the remote PPP entity when transmitting packets to the local PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

pppLinkStatusLocalToRemoteProtocolCompression {pppLinkStatusEntry 10}
Integer Read-only

Indicates whether the local PPP entity will use protocol compression when transmitting packets to the remote PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

- enabled (1) - supports protocol compression
- disabled (2) - no support for protocol compression

pppLinkStatusRemoteToLocalProtocolCompression {pppLinkStatusEntry 11}
Integer Read-only

Indicates whether the remote PPP entity will use Protocol Compression when transmitting packets to the local PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

- enabled (1) - supports protocol compression.
- disabled (2) - no support for protocol compression.

pppLinkStatusLocalToRemoteACCompression {pppLinkStatusEntry 12}
Integer Read-only

Indicates whether the local PPP entity will use Address and Control Compression when transmitting packets to the remote PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

- enabled (1) - supports ACC map compression
- disabled (2) - no support for ACC map compression

pppLinkStatusRemoteToLocalACCompression {pppLinkStatusEntry 13}
Integer Read-only

Indicates whether the remote PPP entity will use address and control compression when transmitting packets to the local PPP entity. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

- enabled (1) - supports ACC map compression
- disabled (2) - no support for ACC map compression

pppLinkStatusTransmitFcsSize {pppLinkStatusEntry 14}
Integer Read-only

The size of the Frame Check Sequence (FCS) in bits that the local node will generate when sending packets to the remote node. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up). 16 bit FCS is the only FCS supported.

pppLinkStatusReceiveFcsSize {pppLinkStatusEntry 15}
Integer Read-only

The size of the Frame Check Sequence (FCS) in bits that the remote node will generate when sending packets to the local node. The value of this object is meaningful only when the link has reached the open state (ifOperStatus is up).

pppLinkConfigTable {pppLink 2}
Not-Accessible

A table containing the LCP configuration parameters for this PPP Link. These variables represent the initial configuration of the PPP Link. The actual values of the parameters may be changed when the Link is brought up via the LCP options negotiation mechanism.

pppLinkConfigEntry {pppLinkConfigTable 1}
Not-Accessible

Configuration information about a particular PPP Link.

pppLinkConfigInitialMRU {pppLinkConfigEntry 1}
Integer Read-Write

The initial Maximum Receive Unit (MRU) that the local PPP entity will advertise to the remote entity. If the value of this variable is 0 then the local PPP entity will not advertise any MRU to the remote entity and the default MRU will be assumed. Changes to this object take effect when the link is next restarted.

Currently, only 8192 is supported and read-only access.

pppLinkConfigReceiveACCMap {pppLinkConfigEntry 2}
Octet String Read-Write

The Asynchronous Control Character Map (ACC) that the local PPP entity requires for use on its receive side. In effect, this is the ACC Map that is required in order to ensure that the local modem will successfully receive all characters. The actual ACC map used on the receive side of the link will be a combination of the local node's pppLinkConfigReceiveACCMap and the remote node's pppLinkConfigTransmitACCMap. Changes to this object take effect when the link is next restarted. The only value supported is (0) no ACC map.

pppLinkConfigTransmitACCMap {pppLinkConfigEntry 3}
Octet String Read-Write

The Asynchronous Control Character Map (ACC) that the local PPP entity requires for use on its transmit side. In effect, this is the ACC Map that is required in order to ensure that all characters can be successfully transmitted through the local modem. The actual ACC map used on the transmit side of the link will be a combination of the local node's pppLinkConfigTransmitACCMap and the remote node's pppLinkConfigReceiveACCMap. Changes to this object take effect when the link is next restarted. The only value supported is (0) no ACC map.

pppLinkConfigMagicNumber {pppLinkConfigEntry 4}
Integer Read-Write

If true (2) then the local node will attempt to perform magic number negotiation with the remote node. If false (1) then this negotiation is not performed. In any event, the local node will comply with any magic number negotiations attempted by the remote node, per the PPP specification. Changes to this object take effect when the link is next restarted.

- True (2) - The ATX will attempt to perform magic number negotiation
- False (1) - will not attempt to perform Magic Number negotiation

pppLinkConfigFcsSize {pppLinkConfigEntry 5}
Integer Read-Write

The size of the FCS, in bits, the local node will attempt to negotiate for use with the remote node. Regardless of the value of this object, the local node will comply with any FCS size negotiations initiated by the remote node, per the PPP specification. Changing this object will have effect when the link is next restarted.

Currently only 16 bit FCS is supported.

pppLqrTable {pppLqr 1}
Not-Accessible

Table containing the LQR parameters and statistics for the local PPP entity.

The ATX always has an empty table.

pppLqrEntry {pppLqrTable 1}
Not-Accessible

LQR information for a particular PPP Link. A PPP Link will have an entry in this table if and only if LQR quality monitoring has been successfully negotiated for the specified link.

pppLqrQuality {pppLqrEntry 1}
Integer Read-only

The current quality of the link as declared by the local PPP entity's Link-Quality Management modules. No effort is made to define good or bad, nor the policy used to determine it. The not-determined value indicates that the entity does not actually evaluate the link's quality. This value is used to distinguish the "determined to be good" case from the "no determination made and presumed to be good" case.

pppLqrInGoodOctets {pppLqrEntry 2}
Counter Read-only

The LQR InGoodOctets counter for this link.

pppLqrLocalPeriod {pppLqrEntry 3}
Integer Read-only

The LQR reporting period, in hundredths of a second that is in effect for the local PPP entity.

pppLqrRemotePeriod {pppLqrEntry 4}
Integer Read-only

The LQR reporting period, in hundredths of a second, that is in effect for the remote PPP entity.

pppLqrOutLQRs {pppLqrEntry 5}
Counter Read-only

The value of the OutLQRs counter on the local node for the link identified by ifIndex.

pppLqrInLQRs {pppLqrEntry 6}
Counter Read-only

The value of the InLQRs counter on the local node for the link identified by ifIndex.

pppLqrConfigTable {pppLqr 2}
Not-Accessible

Table containing the LQR configuration parameters for the local PPP entity.

The ATX always has an empty table.

pppLqrConfigEntry {pppLqrConfigTable 1}
Not-Accessible

LQR configuration information for a particular PPP link.

pppLqrConfigPeriod {pppLqrConfigEntry 1}
Integer Read-Write

The LQR reporting period that the local PPP entity will attempt to negotiate with the remote entity, in units of hundredths of a second. Changes to this object take effect when the link is next restarted.

pppLqrConfigStatus {pppLqrConfigEntry 2}
Integer Read-Write

If enabled (2) then the local node will attempt to perform LQR negotiation with the remote node. If disabled (1) then this negotiation is not performed. In any event, the local node will comply with any magic number negotiations attempted by the remote node, per the PPP specification. Changing this object will have effect when the link is next restarted. Setting this object to the value disabled (1) has the effect of invalidating the corresponding entry in the pppLqrConfigTable object. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use.

pppLqrExtnsTable {pppLqr 3}
Not-Accessible

Table containing additional LQR information for the local PPP entity.

The ATX always has an empty table.

pppLqrExtnsEntry {pppLqrExtnsTable 1}
Not-Accessible

Extended LQR information for a particular PPP Link. Assuming that this group has been implemented, a PPP Link will have an entry in this table if and only if LQR Quality Monitoring has been successfully negotiated for the specified link.

pppLqrExtnsLastReceivedLqrPacket {pppLqrExtnsEntry 1}
Octet String Read-only

This object contains the most recently received LQR packet. The format of the packet is as described in the LQM Protocol specification. All fields of the packet, including the “save” fields, are stored in this object. The LQR packet is stored in network byte order. The LAP-B and PPP headers are not stored in this object; the first four octets of this variable contain the magic number field, the second four octets contain the LastOutLQRs field and so on. The last four octets of this object contain the SaveInOctets field of the LQR packet.

8.2 PPP IP TABLE

pppIp [ppp 3]

pppIpTable {pppIp 1}
Not-Accessible

Table containing the IP parameters and statistics for the local PPP entity.

pppIpEntry {pppIpTable 1}
Not-Accessible

IPCP status information for a particular PPP link.

pppIpOperStatus {pppIpEntry 1}
Integer Read-only

The operational status of the IP network protocol. If the value of this object is up, then the finite state machine for the IP network protocol has reached the opened state.

- Opened (1) - the IPCP protocol is in open state
- Not-opened (2) - the IPCP protocol is in anything but the open state.

pppIpLocalToRemoteCompressionProtocol {pppIpEntry 2}
Integer Read-only

The IP compression protocol that the local PPP-IP entity uses when sending packets to the remote PPP-IP entity. The value of this object is meaningful only when the link has reached the open state (pppIpOperStatus is opened).

- None (1) - no IP compression protocol supported
- Vj-tcp (2) - the Van Jacobsen TCP/IP header compression protocol is supported.

The ATX does not support protocol compression.

pppIpRemoteToLocalCompressionProtocol {pppIpEntry 3}
Integer Read-only

The IP compression protocol that the remote PPP-IP entity uses when sending packets to the local PPP-IP entity. The value of this object is meaningful only when the link has reached the open state (pppIpOperStatus is opened).

- None (1) - no IP compression protocol supported
- Vj-tcp (2) - the Van Jacobsen TCP/IP header compression protocol is supported.

pppIpRemoteMaxSlotId {pppIpEntry 4}
Integer Read-only

The Max-Slot-Id parameter that the remote node has advertised and that is in use on the link. If vj-tcp header compression is not in use on the link then the value of this object shall be 0. The value of this object is meaningful only when the link has reached the open state (pppIpOperStatus is opened).

pppIpLocalMaxSlotId {pppIpEntry 5}
Integer Read-only

The Max-Slot-Id parameter that the local node has advertised and that is in use on the link. If vj-tcp header compression is not in use on the link then the value of this object shall be 0. The value of this object is meaningful only when the link has reached the open state (pppIpOperStatus is opened).

pppIpConfigTable {pppIp 2}
Not-Accessible

Table containing configuration variables for the IPCP for the local PPP entity.

pppIpConfigEntry {pppIpConfigTable 1}
Not-Accessible

IPCP information for a particular PPP link.

pppIpConfigAdminStatus {pppIpConfigEntry 1}
Integer Read-Write

The immediate desired status of the IP network protocol. Setting this object to open (1) will inject an administrative open event into the IP network protocol's finite state machine. Setting this object to close (2) will inject an administrative close event into the IP network protocol's finite state machine. The ATX only supports Open (1).

pppIpConfigCompression {pppIpConfigEntry 2}
Integer Read-Write

If none (1), then the local node will not attempt to negotiate any IP Compression option. Otherwise, the local node will attempt to negotiate compression mode indicated by the enumerated value. Changes to this object take effect when the link is next restarted. The other option Van-Jacobsen TCP/IP header compression (2), is not supported on the ATX.

8.3 PPP BRIDGE SUPPORT

pppBridge {ppp 4}

pppBridgeTable {pppBridge 1}
Not-Accessible

Table containing the parameters and statistics for the local PPP entity that are related to the operation of Bridging over the PPP.

pppBridgeEntry {pppBridgeTable 1}
Not-Accessible

Bridging information for a particular PPP link.

pppBridgeOperStatus {pppBridgeEntry 1}
Integer Read-Only

The operational status of the Bridge network protocol. If the value of this object is opened (1) then the finite state machine for the Bridge network protocol has reached the Opened state. Otherwise, the status is not-opened (2).

pppBridgeLocalToRemoteTinygramCompression {pppBridgeEntry 2}
Integer Read-Only

Indicates whether the local node will perform Tinygram Compression when sending packets to the remote entity. If false (1) then the local entity will not perform Tinygram Compression. If true (2) then the local entity will perform Tinygram Compression. The value of this object is meaningful only when the link has reached the open state (pppBridgeOperStatus is opened).

The unit does not support Tinygram Compression.

pppBridgeRemoteToLocalTinygramCompression {pppBridgeEntry 3}
Integer Read-Only

If false (1) then the remote entity is not expected to perform Tinygram compression and if true (2) then it is. The value of this object is meaningful only when the link has reached the open state (pppBridgeOperStatus is opened).

pppBridgeLocalToRemoteLanId {pppBridgeEntry 4}
Integer Read-Only

Indicates whether the local node will include the LAN Identification field in transmitted packets or not. If false (1) it won't and if true (2) it will. The value of this object is meaningful only when the link has reached the open state (pppBridgeOperStatus is opened).

pppBridgeRemoteToLocalLanId {pppBridgeEntry 5}
Integer Read-Only

Indicates whether the remote node has indicated that it will include the LAN Identification field in transmitted packets or not. If false (1) it won't and if true (2) it will. The value of this object is meaningful only when the link has reached the open state (pppBridgeOperStatus is opened).

pppBridgeConfigTable {pppBridge 2}
Not-Accessible

Table containing the parameters and statistics for the local PPP entity that are related to the operation of Bridging over the PPP.

pppBridgeConfigEntry {pppBridgeConfigTable 1}
Not-Accessible

Bridging configuration information for a particular PPP link.

pppBridgeConfigAdminStatus {pppBridgeConfigEntry 2}
Integer Read-Write

The immediate desired status of the bridging network protocol. Setting this object to open (1) will inject an administrative open event into the bridging network protocol's finite state machine. Setting this object to close (2) will inject an administrative close event into the bridging network protocol's finite state machine.

pppBridgeConfigTinygram {pppBridgeConfigEntry 2}
Integer Read-Write

If false (1) then the local BNCP entity will not initiate the Tinygram Compression Option Negotiation and if true (2) it will.

pppBridgeConfigRingId {pppBridgeConfigEntry 3}
Integer Read-Write

If false (1) then the local PPP Entity will not initiate a Remote Ring Identification Option negotiation and if true (2) it will. This MIB object is relevant only if the interface is for 802.5 Token Ring bridging.

pppBridgeConfigLineId {pppBridgeConfigEntry 4}
Integer Read-Write

If false (1) the local PPP Entity won't initiate a Line Identification Option negotiation and if true (2) it will. This MIB object is relevant only if the interface is for 802.5 Token Ring bridging.

pppBridgeConfigLanId {pppBridgeConfigEntry 5}
Integer Read-Write

If false (1) the local BNCP entity will not initiate the LAN Identification Option Negotiation and if true (2) it will.

pppBridgeMediaTable {pppBridge 3}
Not-Accessible

Table identifying which MAC media types are enabled for the Bridging NCPs.

pppBridgeMediaEntry {pppBridgeMediaTable 1}
Not-Accessible

Status of a specific MAC Type for a specific PPP Link.

pppBridgeMediaMacType {pppBridgeMediaEntry 1}
Integer Read-Only

The MAC type for which this entry in the pppBridgeMediaTable is providing status information. Valid values for this object are defined in Section 6.6 MAC Type Support Selection of RFC1220 (Bridging Point-to-Point Protocol).

The unit only supports a MAC type of FDDI (4).

pppBridgeMediaLocalStatus {pppBridgeMediaEntry 2}
Integer Read-Only

Indicates whether the local PPP Bridging Entity will accept (1) packets of the protocol type identified in pppBridgeMediaMacType on the PPP link identified by ifIndex or not. If this object is accept then any packets or the indicated MAC type will be received and properly process. If this object is don't accept (2) then received packets of the indicated MAC type will not be properly processed.

pppBridgeMediaRemoteStatus {pppBridgeMediaEntry 3}
Integer Read-Only

Indicates whether the local PPP Bridging Entity believes that the remote PPP Bridging Entity will accept (1) packets of the protocol type identified in pppBridgeMediaMacType on the PPP link identified by ifIndex or not.

pppBridgeMediaConfigTable {pppBridge 4}
Not-Accessible

Table identifying which MAC media types are enabled for the bridging NCPs.

pppBridgeMediaConfigEntry {pppBridgeMediaConfigTable 1}
Not-Accessible

Status of a specific MAC type for a specific PPP link.

pppBridgeMediaConfigMacType {pppBridgeMediaConfigEntry 1}
Integer Read-Write

The MAC type for which this entry in the pppBridgeMediaConfigTable is providing status information. Valid values for this object are defined in Section 6.6 MAC Type Support Selection of RFC1220 (Bridging Point-to-Point Protocol).

The unit only supports a MAC type of FDDI (4).

pppBridgeMediaConfigLocalStatus {pppBridgeMediaConfigEntry 2}
Integer Read-Write

Indicates whether the local PPP Bridging Entity should accept (1) packets of the protocol type identified in pppBridgeMediaConfigMacType on the PPP link identified by ifIndex or not. Setting this object to the value don't accept (2) has the affect of invalidating the corresponding entry in the pppBridgeMediaConfigTable object.

It is an implementation-specific matter as to whether the agent removes an invalidated entry form the table. Accordingly, management stations must be prepared to receive tabular information from agents that corresponds to entries not currently in use. Changes to this object take effect when the link is next restarted.

CHAPTER 9
APPLETALK MIB

The unit supports the standard AppleTalk MIB, as defined in RFC 1243. The AppleTalk MIB consists of eight groups:

- LLAP–describes the localTalk Link Access Protocol variables.
- AARRP–describes the AppleTalk Address Resolution Protocol variables.
- ATPORT–describes variables relevant to individual ports on the unit.
- DDP–describes the Datagram Delivery Protocol variables.
- RTMP–describes the Routing Table Maintenance Protocol variables.
- ZIP–describes the Zone information Protocol variables.
- NBP–describes the Name Binding Protocol variables.
- ATECHO–describes the AppleTalk Echo Protocol variables.

For definitions of the variables, refer to the sections that follow.

The MIB tree for reaching the AppleTalk MIB is:

```
1.3.6.1.2.1.
```

appletalk {mib-2 13}

9.1 LLAP GROUP

The LLAP Group describes MIB variables related to the LocalTalk Link Access Protocol.

llap {appletalk 1}

llapTable {llap 1}
Not-Accessible

The list of LLAP entries.

llapEntry {llapTable 1}
Not-Accessible

An LLAP entry containing objects for the LocalTalk Link Access Protocol for a particular LocalTalk interface.

llapIfIndex {llapEntry 1}
Integer Read-Only

The LLAP interface to which this entry pertains. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

llapInPkts {llapEntry 2}
Counter Read-Only

The total number of good packets received on this LocalTalk interface.

llapOutPkts {llapEntry 3}
Counter Read-Only

The total number of packets transmitted on this LocalTalk interface.

llapInNoHandlers {llapEntry 4}
Counter Read-Only

The total number of good packets received on this LocalTalk interface for which there was no protocol handler.

llapInLengthErrors {llapEntry 5}
Counter Read-Only

The total number of packets received on this LocalTalk interface whose actual length did not match the length in the header.

llapInErrors {llapEntry 6}
Counter Read-Only

The total number of packets containing errors received on this LocalTalk interface.

llapCollisions {llapEntry 7}
Counter Read-Only

The total number of collisions assumed on this LocalTalk interface due to the lack of a lapCTS reply.

llapDefers {llapEntry 8}
Counter Read-Only

The total number of times this LocalTalk interface deferred to other packets.

llapNoDataErrors {llapEntry 9}
Counter Read-Only

The total number of times this LocalTalk interface received a lapRTS packet and expected a data packet, but did not receive any data packet.

llapRandomCTSErrors {llapEntry 10}
Counter Read-Only

The total number of times this LocalTalk interface received a lapCTS packet that was not solicited by a lapRTS packet.

llapFCSErrors {llapEntry 11}
Counter Read-Only

The total number of times this LocalTalk interface received a packet with an FCS (Frame Check Sequence) error.

9.2 AARP GROUP

The AARP group describes MIB variables related to the AppleTalk Address Resolution Protocol.

aarp {appletalk 2}

aarpTable {aarp 1}
Not-Accessible

The AppleTalk Address Translation Table contains an equivalence of AppleTalk Network Addresses to the link layer physical address.

aarpEntry {aarpTable 1}
Not-Accessible

Each entry contains one AppleTalk Network Address to physical address equivalence.

aarpIfIndex {aarpEntry 1}
Integer Read-Only

The interface on which this entry's equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

aarpPhysAddress {aarpEntry 2}
Octet String Read-Only

The media-dependant physical address.

aarpNetAddress {aarpEntry 3}
Octet String Read-Only

The AppleTalk Network Address corresponding to the media-dependant physical address.

9.3 ATPORT GROUP

The ATPort group describes MIB variables related to individual ports on the unit.

atport {appletalk 3}

atportTable {atport 1}
Not-Accessible

A list of AppleTalk ports for this entity.

atportEntry {atportTable 1}
Not-Accessible

The description of one of the AppleTalk ports on this entity.

atportIndex {atportEntry 1}
Integer Read-Only

A unique value for each AppleTalk port. Its value is between 1 and the total number of AppleTalk ports. The value for each port must remain constant at least form the re-initialization of the entity's network management system to the next re-initialization.

atportDescr {atportEntry 2}
Octet String Read-Only

A text string containing information about the port. This string is intended for presentation to a human; it must not contain anything but printable ASCII characters.

atportType {atportEntry 3}
Integer Read-Write

The type of port, distinguished by the protocol immediately below DDP in the protocol stack.

atportNetStart {atportEntry 4}
Octet String Read-Write

The first AppleTalk network address in the range configured for this port. This is a two octet DDP network address in network byte order.

atportNetEnd {atportEntry 5}
Octet String Read-Write

The last AppleTalk network address in the range configured for this port. This is a two octet DDP network address in network byte order. If the network to which this AppleTalk port is connected is a Phase 1 network or a non-extended network, the value for atportNetEnd shall be two octets of zero.

atportNetAddress {atportEntry 6}
Octet String Read-Write

The AppleTalk network address configured for this port.

atportStatus {atportEntry 7}
Integer Read-Write

The configuration status of this port. Setting this object to the value invalid (4) has the effect of invalidating the corresponding entry in the atportTable. That is, it effectively disassociates the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive from agents tabular information corresponding to entries not currently in use. Proper interpretation of such entries requires examination of the relevant atportStatus object.

The values for atportStatus include:

- operational (1)
- unconfigured (2)
- off (3)
- invalid (4)

atportNetConfig {atportEntry 8}
Integer Read-Only

The configuration status of this port.

atportZoneConfig {atportEntry 9}
Integer Read-Only

The configuration status of the zone information for this port.

atportZone {atportEntry 10}
Octet String Read-Write

The zone name configured for this AppleTalk port.

atportIfIndex {atportEntry 11}
Integer Read-Write

The physical interface associated with this AppleTalk port. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

9.4 DDP GROUP

The DDP group describes MIB variables related to the Datagram Delivery Protocol.

ddp {appletalk 4}

ddpOutRequests {ddp 1}
Counter Read-Only

The total number of DDP datagrams which were supplied to DDP by local DDP clients in requests for transmission. Note that this counter does not include any datagrams counted in ddpForwRequests.

ddpOutShorts {ddp 2}
Counter Read-Only

The total number of short DDP datagrams which were transmitted from this entity.

ddpOutLongs {ddp 3}
Counter Read-Only

The total number of long DDP datagrams which were transmitted from this entity.

ddpInReceives {ddp 4}
Counter Read-Only

The total number of input datagrams received by DDP, including those received in error.

ddpForwRequests {ddp 5}
Counter Read-Only

The number of input datagrams for which this entity was not their final DDP destination, as a result of which an attempt was made to find a route to forward them to that final destination.

ddpInLocalDatagrams {ddp 6}
Counter Read-Only

The total number of input DDP datagrams for which this entity was their final DDP destination.

ddpNoProtocolHandlers {ddp 7}
Counter Read-Only

The total number of DDP datagrams addressed to this entity that were addressed to an upper layer protocol for which no protocol handler existed.

ddpOutNoRoutes {ddp 8}
Counter Read-Only

The total number of DDP datagrams dropped because a route could not be found to their final destination.

ddpTooShortErrors {ddp 9}
Counter Read-Only

The total number of input DDP datagrams dropped because the received data length was greater than the data length specified in the DDP header or the received data length was less than the length of the expected DDP header.

ddpTooLongErrors {ddp 10}
Counter Read-Only

The total number of input DDP datagrams dropped because the received data length was greater than the data length specified in the DDP header or because they exceeded the maximum DDP datagram size.

ddpBroadcastErrors {ddp 11}
Counter Read-Only

The total number of input DDP datagrams dropped because this entity was not their final destination and they were addressed to the link level broadcast.

ddpShortDDPErrors {ddp 12}
Counter Read-Only

The total number of input DDP datagrams dropped because this entity was not their final destination and their type was short DDP.

ddpHopCountErrors {ddp 13}
Counter Read-Only

The total number of input DDP datagrams dropped because this entity was not their final destination and their hop count would exceed 15.

ddpChecksumErrors {ddp 14}
Counter Read-Only

The total number of input DDP datagrams dropped because of a checksum error.

9.5 RTMP GROUP

The RTMP group describes MIB variables related to the Routing Table Maintenance Protocol.

rtmp {appletalk 5}

rtmpTable {rtmp 1}

A list of Routing Table Maintenance Protocol entries for this entity.

rtmpEntry {rtmpTable 1}
RtmpEntry

The route entry to a particular network range.

rtmpRangeStart {rtmpEntry 1}
Octet StringRead-Write

The first DDP network address in the network range to which this routing entry pertains. This is a two octet DDP network address in network byte order.

rtmpRangeEnd {rtmpEntry 2}
Octet String Read-Write

The last DDP network address in the network range to which this routing entry pertains. This is a two octet DDP network address in network byte order. If the network to which this routing entry pertains is a Phase 1 network or a non-extended network, the value for rtmpRangeEnd shall be two octets of zero.

rtmpNextHop {rtmpEntry 3}
Octet String Read-Write

The next hop in the route to this entry's destination network. If the type of this route is Appletalk, this address takes the same form as DdpAddress.

rtmpType {rtmpEntry 4}
Integer Read-Write

The type of network over which this route points.

rtmpPort {rtmpEntry 5}
Integer Read-Write

The index of the AppleTalk port over which this route points.

rtmpHops {rtmpEntry 6}
Integer Read-Write

The number of hops required to reach the destination network to which this routing entry pertains.

rtmpState {rtmpEntry 7}
Integer Read-Write

The status of the information contained in this route entry. Setting this object to the value bad (4) has the effect of invalidating the corresponding entry in the rtmpTable. That is, it effectively disassociates the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive from agents tabular information corresponding to entries not currently in use. Proper interpretation of such entries requires examination of the relevant rtmpState object.

The values for rtmpState include:

- good (1)
- suspect (2)
- goingBad (3)
- bad (4)–may be removed from table.

9.6 ZIP GROUP

The zip group describes MIB variables related to the Zone Information Protocol.

zip {appletalk 7}

zipTable {zip 1}
Not-Accessible

The table of zone information for reachable AppleTalk networks.

zipEntry {zipTable 1}
Not-Accessible

An entry of zone information for a particular zone and network combination.

zipZoneName {zipEntry 1}
Octet String Read-Write

The ASCII zone name of this entry.

zipZoneIndex {zipEntry 2}
Integer Read-Only

An integer that is unique to the zipZoneName that is present in this entry. For any given zone name, every zipEntry that has an equal zone name will have the same zipZoneIndex.

zipZoneNetStart {zipEntry 3}
Octet String Read-Write

The network that starts the range for this entry. This address is a two octet DDP network address in network byte order.

zipZoneNetEnd {zipEntry 4}
Octet String Read-Write

The network that ends the range for this entry. This address is a two octet DDP network address in network byte order. If the network to which this zip entry pertains is a Phase 1 network or a non-extended network, the value for zipZoneNetEnd shall be two bytes of zero.

zipZoneState {zipEntry 5}
Integer Read-Write

The state of this zip entry. Setting this object to the value invalid (2) has the effect of invalidating the corresponding entry in the zipTable. That is, it effectively disassociates the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive from agents tabular information corresponding to entries not currently in use. Proper interpretation of such entries requires examination of the relevant zipZoneState object.

The zipZoneState values include:

- valid (1)
- invalid (2)

9.7 NBP GROUP

The NBP group describes MIB variables related to the Name Binding Protocol.

nbp {appletalk 8}

nbpTable {nbp 1}
Not-Accessible

The table of NBP services registered on this entity.

nbpEntry {nbpTable 1}
Not-Accessible

The description of an NBP service registered on this entity.

nbpIndex {nbpEntry 1}
Integer Read-Only

The index of this NBP entry. This value ranges from 1 to the number of NBP entries currently registered on this entity.

nbpObject {nbpEntry 2}
Octet String Read-Only

The name of the service described by this entity.

nbpType {nbpEntry 3}
Octet String Read-Write

The type of the service described by this entity.

nbpZone {nbpEntry 4}
Octet String Read-Write

The zone the service described by this entity is registered in.

nbpState {nbpEntry 5}
Integer Read-Write

The state of this NBP entry. Setting this object to the value invalid (2) has the effect of invalidating the corresponding entry in the nbpTable. That is, it effectively disassociates the mapping identified with said entry. It is an implementation-specific matter as to whether the agent removes an invalidated entry from the table. Accordingly, management stations must be prepared to receive from agents tabular information corresponding to entries not currently in use. Proper interpretation of such entries requires examination of the relevant nbpState object.

9.8 ATECHO GROUP

The ATEcho group describes MIB variables related to the AppleTalk Echo Protocol.

atecho {appletalk 9}

atechoRequests {atecho 1}
Counter Read-Only

The number of AppleTalk echo requests received.

atechoReplies {atecho 2}
Counter Read-Only

The number of AppleTalk echo replies sent.

CHAPTER 10
CABLETRON MIB

The Cabletron MIB {enterprise 97} is divided into several groups of parameters. The individual groups are described in the subsections below.

The MIB tree prefix for reaching the private enterprise Cabletron MIB-II is:

```
1.3.6.1.4.1.97.
```

sigma {enterprise 97}

10.1 SYSTEM GROUP

sys {sigma 1}

The System Group contains those parameters which are likely to be applicable for all of the products in the ATX product lines. A detailed description of the parameters follows.

sysID {sys 1}
Integer Read-Only

An integer which serves to identify the type of hardware platform. This document describes the ATX bridge/router, which uses the value (1) for sysID.

sysReset {sys 2}
TimeTicks Read-Write

The time, in centiseconds, before the ATX should reboot. A GetResponse value of zero indicates that no reboot time has been specified.

sysTrapAck {sys 3}
Boolean Read-Write

Whether the NMS will acknowledge the enterprise specific traps.

sysTrapTime {sys 4}
TimeTicks Read-Write

The time, in centiseconds, which should expire between the sending of traps. This value is used when no ICMP source quench datagrams have been received. [Receiving ICMP source quench datagrams will cause the unit to use a time period greater than sysTrapTime]. The unit uses sysTrapTime if any SNMP requests from the NMS has been received within the previous sixty seconds. If no requests have been received, then the unit reverts to its default of a fifteen second wait between the sending of traps.

sysTrapRetry {sys 5}
TimeTicks Read-Write

The time, in centiseconds, which should expire before an unacknowledged Trap is re-transmitted.

sysTrapPort {sys 6}
Integer Read-Write

Zero, or the UDP port number to which a second copy of SNMP traps should be sent. Valid values are 0 through 65535.

10.2 RESERVED GROUP

{sigma 2} Not-Applicable

This subtree holds information which is not relevant for the unit.

10.3 ATX GROUP

ecs-1 {sigma 3}

This subtree holds all of the information for the unit. These attributes are sheltered within a single tree in an attempt to avoid future compatibility problems. Since the unit is a multi-media unit, packets are converted from the different media into a standard canonical format, and parameters within this subtree may refer to the canonical format packet. The exact translation procedure is defined by RFC 1188 and RFC 1042, except for the encapsulation of Ethernet AppleTalk packets which uses a Protocol ID of 00-00-F8 instead of all zeroes. For further information, refer to the RFCs. The canonical format is also diagrammed in Appendix B of the *ATX User Guide*.

10.3.1 Hardware Configuration Group

hw {ecs-1 1}

This attribute group supplies information which describes the configuration and status of this particular ATX.

hwNumber {hw 1}
Integer Read-Only

The number of slots available for hardware modules, regardless of whether or not the slots are currently populated. This number is fixed as 7.

hwSlotTable {hw 2}
Not-Accessible

A list of the most recent occupants of the slots; one per slot.

hwSysBus {hw 3}
Integer Read-Only

hwPpeType {hw 4}
Integer Read-Only

hwSysProcessor {hw 5}
Integer Read-Only

hwEntry {hwSlotTable 1}
Not-Accessible

A set of objects for each slot. The individual components are described below.

hwIndex {hwEntry 1}
Integer Read-Only

The slot number, beginning with 1 for the top-most slot. This number always matches the instance of the hwEntry. For example, {hwEindex 3} contains the value “3”.

hwType {hwEntry 2}
Integer Read-Only

The module type of the slot's most recent occupant since the unit was started. The values for hwType correspond to the values for hwPortType (defined later), and are defined as follows:

- vacant (1) - slot is not occupied
- unknown (2) - reserved
- packet-processing-engine (3) - Packet Processing Engine
- turbo (4) - Turbo Processor
- csma-iom (5) - Ethernet/802.3 IOM
- hssi-iom (6) - HSSI IOM
- tpr-iom (7) - Token Passing Ring IOM
- fddi-iom (8) - FDDI IOM
- ifddi-iom (9) - Intelligent FDDI IOM
- ttpr-iom (10) - Turbo Token Passing Ring IOM, e.g., the QTIOM

hwUseMod {hwEntry 3}
Integer Read-Write

Whether to use this module, not applicable for the Packet Processing Engine. The value of hwUseMod is not saved if the unit reboots (see rebootUseMod). If the module fails diagnostics, it will not be used, regardless of hwUseMod. Possible values include:

- reset (1) - do not use the module (keep it in reset) -before a module is hot swapped, it must be put into this state.
- run-diagnostics (2) - do not use the module (run diagnostics; after diagnostics complete, keep the module in reset and automatically set hwUseMod to 1)
- run (3) - use the module

hwDefType {hwEntry 4}
Integer Read-Only

The module type defined for the slot. Certain pre-defined relationships exist. Slot 1 always contains the Packet Processing Engine board. Slots 2-6 are either unoccupied, or contain IOM boards. Slot 7 is either unoccupied or contains the Turbo Processor board. If hwDefType does not match hwType, then the module will not be used, regardless of the value of hwUseMod. Note that hwDefType can only be changed by rebooting the unit (see the "reboot" group).

hwDiagStatus {hwEntry 5}
Integer Read-Only

Results of diagnostics, when diagnostics were last performed on the module (whenever a module is released from reset, which is usually at power-up). Possible values include:

- diag-failed (1) - diagnostics failed
- diag-not-present (2) - module not present (or possibly, diagnostics still running, or the module's diagnostics are totally non-responsive)
- diag-passed (3) - diagnostics passed

hwInuse {hwEntry 6}
Boolean Read-Only

Whether the module is currently executing its operational software.

hwDiagCode {hwEntry 7}
Octet String Read-Only

Additional diagnostics information concerning the slot's module. The meaning of hwDiagCode depends upon the slot's module type described by the hwType parameter.

For the Packet Processing Engine board, any combination of the following errors may be combined; however, any error (except where noted otherwise) will prevent the unit from booting:

0x00001	VDRAM and/or Processor error
0x00002	Shared Ram error
0x00004	Rom Checksum error
0x00008	Configuration EEPROM Checksum error
0x00010	Flash EEPROM Bank 0 Checksum error (if Bank 1 is valid, then the unit will still boot)

0x00020	ID EEPROM error
0x00040	UART error (the unit will still boot)
0x00080	Cam error
0x00100	Interrupt Control error
0x00200	Watchdog Timeout error
0x00400	Voltage Monitor error
0x00800	Temperature Sensor error
0x01000	Flash EEPROM Bank 1 Checksum error (if Bank 0 is valid, then the unit will still boot)
0x02000	reserved
0x04000	Semaphore error
0x08000	reserved
0x10000	Packet Processing Engine Arbitration error

For the Turbo processor board, any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	VDRAM and/or Processor error
0x00002	Shared Ram error
0x00004	Rom Checksum error
0x00008	reserved
0x00010	reserved
0x00020	reserved
0x00040	UART error
0x00080	Cam error
0x00100	Interrupt Control error
0x00200	reserved
0x00400	reserved
0x00800	reserved
0x01000	reserved
0x02000	reserved
0x04000	reserved
0x08000	reserved
0x10000	Turbo Arbitration error

For the FDDI IOM board (FIOM), any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	RBC error
0x00002	DPC error
0x00004	FORMAC error
0x00008	ENDEC error
0x00010	FDDI Buffer Memory error
0x00020	RDMAC error
0x00040	Timers error
0x00080	Cam error
0x00100	ID EEPROM error
0x00200	Temperature Sensor error
0x00400	Source Strip Logic error
0x00800	FIOM Arbitration error
0x01000	Data Access Exception error

For the intelligent FDDI IOM board (IFIOM), any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	Local RAM error
0x00002	Shared RAM error
0x00004	Descriptor RAM error
0x00008	ROM Checksum error
0x00010	ID EEPROM Checksum error
0x00020	UART error
0x00040	Temperature Sensor error
0x00080	MACSI Control Bus Interface error
0x00100	FDDI Loopback error
0x00200	CAM error
0x00400	LANCAM error
0x00800	Semaphore error
0x01000	Arbitration error
0x02000	Features “COMPAT” error

For the Ethernet/802.3 IOM boards (including the single-port Ethernet board, the four port Ethernet board, and the concentrator Ethernet boards), any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	Local SRAM error
0x00002	Shared Ram error
0x00004	Rom Checksum error
0x00008	Sonic error (1st Sonic, if multiple Sonics are present)
0x00010	Ethernet IOM Arbitration error
0x00020	ID EEPROM Checksum error
0x00040	UART error
0x00080	Semaphore error
0x00100	CAM error (only present on the four port board)
0x00200	Sonic error (2nd Sonic, only if present)
0x00400	Sonic error (3rd Sonic, only if present)
0x00800	Sonic error (4th Sonic, only if present)

For the Token Ring/802.5 IOM boards (including the single port Token Ring board and the four port Turbo Token Ring board), any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	Local SRAM error
0x00002	Shared Ram error
0x00004	Rom Checksum error
0x00008	TMS380 error (1st TMS380 if multiple TMS380's are present)
0x00010	Token Ring IOM Arbitration error
0x00020	ID EEPROM Checksum error
0x00040	UART error
0x00080	Semaphore error
0x00100	CAM error
0x00200	TMS380 error (2nd TMS380, only if present)
0x00400	TMS380 error (3rd TMS380, only if present)
0x00800	TMS380 error (4th TMS380, only if present)
0x01000	Am 29030 error

For the HSSI IOM board, any combination of the following errors may be combined; however, any error will prevent the board from being used:

0x00001	Local SRAM error
0x00002	Shared Ram error
0x00004	Rom Checksum error
0x00008	DMA Circuitry error
0x00010	HSSI IOM Arbitration error
0x00020	ID EEPROM Checksum error
0x00040	UART error
0x00080	Semaphore error
0x00100	CAM error
0x00200	Compatibility (with the PPE) error

hwManufData {hwEntry 8}
DisplayString Read-Only

This is a 24-byte array which contains the part number, serial number, and hardware revision level of this unit. The Turbo's hwManufData will be all blanks since it is manufactured as part of the Packet Processing Engine module.

hwPortType {hwEntry 9}
Integer Read-Only

A 4 byte array, each element represents the port types belonging to this module. The Packet Processing Engine's ports are fixed as "uart" for the first port, and "none" for the other three ports. The Turbo's ports are fixed as "none" for all ports. The following port types are supported:

- none (0) - there is no port
- csma (1) - Ethernet or 802.3 port
- fddi-das (2) - FDDI DAS port
- fddi-sas (3) - FDDI SAS port
- tpr (4) - Token Passing Ring port
- wan (5) - Wide Area Network port
- uart (6) - UART port

hwPortStatus {hwEntry 10}
Octet String Read-Only

A 4 byte array, where each element represents the status of the port belonging to this module . Values of 01 (status-good) or hex ff (status-bad) are supported. For the Packet Processing Engine, the first byte is the status of the UART port.

hwUsePort {hwEntry 11}
Obsolete

hwDefPortType {hwEntry 12}
Octet String Read-Only

A 4 byte array, where each element represents the port types defined for the module. If hwDefPortType does not match hwPortType, then the port will not be used. Note that hwDefPortType can only be changed by rebooting the unit.

hwAddr1 {hwEntry 13}
MacAddress Read-Only

The unique hard-wired 48 bit MAC address for the module's 1st port. For the Packet Processing Engine, it's a unique MAC address to be used as part of the unit's Spanning Tree Bridge Identifier. Not applicable for the Turbo.

hwAddr2 {hwEntry 14}
MacAddress Read-Only

A unique hard-wired 48 bit MAC address for the modules's 2nd port, only applicable for IOMs.

hwAddr3 {hwEntry 15}
MacAddress Read-Only

A unique hard-wired 48 bit MAC address for the modules's 3rd port, only applicable for IOMs.

hwAddr4 {hwEntry 16}
MacAddress Read-Only

A unique hard-wired 48 bit MAC address for the modules's 4th port, only applicable for IOMs.

hwTempOk {hwEntry 17}
Boolean Read-Only

The temperature condition of the module, where True indicates temperature okay, and False indicates temperature too hot.

hwFirstPort {hwEntry 18}
Integer Read-Only

The IfIndex number of the first port which resides within this module.

hwFatalErr {hwEntry 19}
Octet String Read-Only

A 32 byte array which contains information about the cause of the module's last unexpected self-reset, since the unit rebooted. For the Packet Processing Engine and Turbo, this will always be all zeroes, since failure of such a module will cause the unit to reboot. For the other modules, this field either contains all zeroes (if the module has not self-reset), or it contains values according to the format of configFatalErr, with a first byte of either "Watchdog Timeout" (1) or "Fatal Error" (3).

hwRptrPorts {hwEntry 20}
Octet String Read-Only

A 4 byte array, where each element represents the repeater capability of the port belonging to this module. The element value is zero if the port is not present or does not have any repeater capability; otherwise, the value indicates the number of repeater ports at that port (see rptrBasGroupNumOfPorts).

hwPortSubType {hwEntry 21}
Octet String Read-Only

A 4 byte array, where each element represents additional port type information. The following values have been defined:

0x01	CSMA/CD AUI
0x02	CSMA/CD TP repeater with 6 TP ports; limited repeater management is available (not currently available as a product)
0x03	CSMA/CD TP repeater with 6 TP ports, full repeater management is available (not currently available as a product)
0x04	CSMA/CD TP with 12 TP ports; full repeater management is available (not currently available as a product)
0x05	CSMA/CD TP with 12 TP and 1AUI port; full repeater management is available (not currently available as a product)
0x06	CSMA/CD TP with 24 TP ports; full repeater management is available
0x07	CSMA/CD TP with 24 TP and 1AUI; full repeater management is available (not currently available as a product)

0x08	CSMA/CD TP port
0x09	CSMA/CD BNC port
0x0a	CSMA/CD FO port
0x0b	CSMA/CD TP port with dual RJ-45 jacks (one normal, one crossover).
0x0c	CSMA/CD with internal digitized Ethernet
0x20	Token Ring port with a shielded connector
0x21	Token Ring port with an unshielded connector
0x22	Token Ring with dual connectors (one shielded and one unshielded)
0x30	HSSI port with dual jacks (one HSSI; one RS-449 for lower speeds).
0x40	FDDI (DAS) port with Single Mode fiber
0x41	FDDI (DAS) port with Multimode fiber
0x42	FDDI (DAS) port with Single Mode fiber on Mic A and Multimode fiber on Mic B
0x43	FDDI (DAS) port with Multimode fiber on Mic A and Single Mode fiber on Mic B
0xff	Port not present, or no additional information is available.

hwAddr5 {hwEntry 22}
Octet String Read-Only

hwAddr6 {hwEntry 23}
Octet String Read-Only

hwAddr7 {hwEntry 24}
Octet String Read-Only

hwAddr8 {hwEntry 25}
Octet String Read-Only

10.3.2 Software Configuration Group

sw {ecs-1 2}

This attribute group supplies information which describes the software configuration of this particular ATX.

swNumber {sw 1}
Integer Read-Only

The number of file sets. This number is fixed as 2.

swFilesetTable {sw 2}
Not-Accessible

A list of the software file sets stored in the ATX. The first file set refers to the file set that was used when the ATX last booted. The second file set refers to the file set that will be used when the ATX reboots or reloads any hardware module. [Since IOM hardware modules may be re-started after the unit has booted, the software which is executing on an IOM may be different than the software that was used when the unit booted.].

swFileset {swFilesetTable 1}
Not-Accessible

A set of objects for each file set. The individual components are described below.

swIndex {swFileset 1}
Integer Read-Only

The file set number, beginning with 1 for the first file set. This number always matches the instance of the swFiles. For example, {swIndex 2} contains the value “2”.

swValid {swFileset 2}
Obsolete

swDesc {swFileset 3}
Octet String Read-Only

The description (0-32 bytes) of the software file set.

swCount {swFileset 4}
Integer Read-Only

The number of files in the file set. This number may range from 1 to 255.

swTypes {swFileset 5}
Octet String Read-Only

The types of files within the file set. The size of swFileTypes may be determined by swCount, since 1 octet is required for each file. The possible file types are

- none (0) - File not present
- ppe-turbo (1) - Packet Processing Engine and/or Turbo Code
- ppe-only (2) - Packet Processing Engine Code
- turbo-only (3) - Turbo Code
- eiom (4) - EIOM Code (for any Ethernet/802.3 IOM)
- hiom (5) - HSSI IOM Code
- tiom (6) - Token Ring IOM Code
- fiom (7) - FIOM Code
- ifiom (8) - Intelligent FIOM Code
- ttiom (9) - Turbo Token Ring IOM Code

swSizes {swFileset 6}
Octet String Read-Only

An array (4 octets per file), containing the size of each of the files. Each size is encoded as a series of 4 bytes, which should be converted into a 32 bit integer.

swStarts {swFileset 7}
Octet String Read-Only

An array (4 octets per file), containing the software's execution starting address of each of the files. Each address is encoded as a series of 4 bytes, which should be converted into a 32 bit integer.

swBases {swFileset 8}
Octet String Read-Only

An array (4 octets per file), containing the software's base loading address of each of the files. Each address is encoded as a series of 4 bytes, which should be converted into a 32 bit integer.

swFlashBank {swFileset 9}
Integer Read-Only

The bank number where the software file set resides. The possible values are:

- first-bank (1)
- second-bank (2)

10.3.3 Unit Administration Group

admin {ecs-1 3}

This group contains the attributes for configuring the system level control of the entire ATX and the overall ATX status.

10.3.3.1 Configuration Status Group

config {admin 1}

This group of variables is used by the NMS to learn and modify the configuration status of the entire unit.

configFatalErr {config 1}
Octet String Read-Only

This is a 32 byte array which contains information about the cause of the previous system reset. This will describe the circumstances which forced the system software to perform a reboot. The values in the first byte have the following meanings:

- (0) - Power failure (remaining 31 bytes are not applicable).
- (1) - Watchdog Timeout (next byte contains the hwIndex of the module which stopped responding; remaining 30 bytes are not applicable).
- (2) - NMS Requested Shutdown (next byte contains the hwIndex of the Packet Processing Engine; next two bytes are not applicable; next four bytes contain the NMS's IP address; remaining 24 bytes are not applicable).
- (3-255) - Fatal Error (next byte contains the hwIndex of the module which encountered the error; remaining 30 bytes contain additional information).

configAnyPass {config 2}
Display String Read-Write

The authentication password (0-24 bytes) which must match the community name in an SNMP PDU, in order for the operation to be performed. A zero length password indicates that any community name is acceptable.

configGetPass {config 3}
Display String Read-Write

The authentication password (0-24 bytes) which must match the community name in an SNMP Get PDU (except for Debug or password parameters). A zero length password indicates that any community name is acceptable. Alternatively, it is acceptable if the community name matches configAnyPass, defined above.[1]

configNMSAddr {config 4}
IP Address Read-Write

The address of the NMS to which Trap PDUs are to be sent. A value of zero indicates that Trap PDUs should be sent to the NMS last heard from.

configFunctions {config 5}
Integer Read-Only

A bit field which describes the software functions supported by the unit. The possible values (or combinations thereof) are:

0x0001	NMS - communicates with NMS
0x0002	TBRIDGE - Transparent IEEE 802.1d MAC bridging (if SBRIDGE is also set, then SRT MAC bridging is supported)
0x0004	SBRIDGE - Source Routing MAC bridging
0x0008	MESH - reserved (originally intended for Cabletron Mesh Algorithm bridging over WAN ports)
0x0010	IPROUTER - IP Routing

1. Certain SetRequest PDUs are used for read operations or for acknowledging alarms, and using configGetPass for the community name is sufficient. Those special conditions consist of SetRequest PDUs for the following fields: addrFlags, addrMAC, addrPort, addrOperation (for read-random, read-next, and read-block only), addrIndex, addrIndexMesh, addrNext, addrAge, addrRxPkts, addrRxChars, addrRxMultiPkts, addrRxFwdPkts, addrTxPkts, addrTxChars, addrBlockSize, sfddiSrfConditions, sfddiSmtConditions, addrBlock and addrRptrPort.

0x0020	NO_BPDU - if enabled, prevents BPDUs from being transmitted and keeps ports in Forwarding state (this is to provide interoperability with non-compliant IEEE 802.1d implementations; however, if the physical topology has any loops, then LAN segments will be flooded with duplicates of packets).
0x0040	RIP - if IPROUTER is set, then this enables the Routing Information Protocol (RFC 1058)
0x0100	PROXY - if IPROUTER is set, then this enables Proxy ARP (RFC 1027)
0x0400	IPX - IPX routing is supported
0x0800	ARP
0x4000	BOOTP
0x8000	FRAME_SWITCH
0x10000	TRUNKING
0x20000	IPM (multicast)
0x40000	APPLETALK

configPowerAc1 {config 6}
Boolean Read-Only

Whether the first power supply has AC input.

configPowerAc2 {config 7}
Boolean Read-Only

Whether the second power supply has AC input.

configPowerDc1 {config 8}
Boolean Read-Only

Whether the first power supply provides DC voltage.

configPowerDc2 {config 9}
Boolean Read-Only

Whether the second power supply provides DC voltage.

configPowerPresent1 {config 10}
Boolean Read-Only

Whether the first power supply is present.

configPowerPresent2 {config 11}
Boolean Read-Only

Whether the second power supply is present.

configAlarmDynamic {config 12}
Boolean Read-Write

Whether changes in dynamic parameters (except for the dynamically learned addresses) should cause enterprise specific trap PDUs to be generated.

configAlarmAddresses {config 13}
Boolean Read-Write

Whether changes in the dynamically learned addresses should cause enterprise specific trap PDUs to be generated.

configStarageFailure {config 14}
Boolean Read-Only

Whether the unit's storage facility for its configuration parameters has failed (the unit may not be able to reboot after such a failure).

configAuthenticationFailure {config 15}
IPAddress Read-Only

All nulls, or the IP source address within the last SNMP PDU which caused an SNMP authentication failure.

configFddiPriority {config 16}
Integer Read-Write

Priority to use when forwarding packets to any FDDI port from any non-FDDI port.

configTprPriority {config 17}
Integer Read-Write

Priority to use when forwarding packets to any Token Ring port from and non-Token Ring port.

configDumpModule {config 19}
Integer Read-Write

Flag to select the module to dump on a crash. If set to a value of 1 to 7 the relevant module's memory will be sent to the UART port on detecting a fatal error situation. A value of zero will disable the dump option. A value of 1 selects the ME, 7 selects the TURBO, and 2-6 select IOM's. See configDumpStart and configDumpEnd.

configDumpStart {config 20}
Integer Read-Write

Starting address of memory to dump upon detecting a fatal error. See configDumpModule and configDumpEnd.

configDumpEnd {config 21}
Integer Read-Write

Ending address of memory to dump upon detecting a fatal error. See configDumpModule and configDumpStart.

10.3.3.2 LMA Configuration Group

lma {admin 2}

This group of variables is used by the NMS to learn and modify the configuration status of the unit's Local Management Agent.

lmaAllAddr {lma 1}
MacAddress Read-Write

The IEEE all LANs Bridge Management Group Address 01-80-c2-00-00-10. Changing lmaAllAddr will not take effect until the unit reboots.

lmaAnyAddr {lma 2}
MacAddress Read-Write

A spare LMA address, which is always all zeroes. Changing ImaAllAny will not take effect until the unit reboots.

10.3.3.3 PPE Configuration Group

ppe {admin 3}

This attribute group permits customization and monitoring of the Packet Processing Engine's operational parameters.

ppeLrgUxRxCnt {ppe 1}
Integer Read-Write

The number of large data buffers to allocate for the PPE to create packets. This value may be changed automatically by the unit as a result of enabling IPX routing. The unit will calculate the minimum number of such buffers that are needed for the number of configured ports and may adjust this count upward if needed.

ppeSmUxRxCnt {ppe 2}
Integer Read-Write

The number of small data buffers to allocate for the PPE to create packets.

ppeUxTxCnt {ppe 3}
Integer Read-Write

The maximum number of received packets destined to the unit itself (versus those received packets to be forwarded) which can be stored, until the PPE is able to process them.

ppeSmlBuffSize {ppe 4}
Integer Read-Write

The size (in bytes) of the PPE's small data buffers.

ppeBridgingMemory {ppe5}
Integer Read-Only

The percentage (1-99) of the PPE's memory that is allocated for MAC bridging. The PPE's remaining memory will be used for communicating with the NMS and for routing. See rebootBridgingMemory. Operationally, a value between 70 and 99 percent will be treated as 70 percent. This allows for approximately 8000 learned addresses (bridging) and either 1600 TCP/IP routes or 1200 IPX RIP/SAP pairs. To increase the space available for routing, set rebootBridgingMemory to a value less than 70 percent.

ppeExtendStats {ppe 6}
Boolean Read-Write

Whether the PPE should accumulate extended statistics. Extended statistics are tallies, such as byte counts, which are not essential for everyday ATX management, and which may have a slightly detrimental effect on performance.

ppeBAddrLimit {ppe 7}
Integer Read-Write

The maximum number of addresses that the PPE should learn for bridging purposes. The remaining address entries will be used for routing related addresses.

ppeTxCongests {ppe 8}
Counter Read-Only

Number of packets not received due to internal buffer congestion.

ppeArpEntries {ppe 9}
Counter Read-Only

Number of ARP entries for all interfaces.

ppeArpStatics {ppe 10}
Counter Read-Only

Number of statistically defined ARP entries for all interfaces.

ppeArpOverflows {ppe 11}
Counter Read-Only

Number of times a ARP entry could not be learned due to insufficient address table space.

ppeIpEntries {ppe 12}
Counter Read-Only

Number of IP Routing Database entries.

ppeIpStatics {ppe 13}
Counter Read-Only

Number of statically defined IP Routing Database entries.

ppeStaticPreference {ppe 14}
Integer Read-Write

The value to assign to ipRouteMetric2, when adding a statically-defined entry to the IP Routing Table. The route preference value for a static entry is 1. Note that for route preferences the smaller the value, the higher the preference.

ppeOspfPreference {ppe 15}
Integer Read-Write

The value to assign to ipRouteMetric2, when adding an OSPF-learned entry to the IP Routing Table. The route preference value for an OSPF-learned entry is 8. Note that for route preferences the smaller the value, the higher the preference.

ppeRipPreference {ppe 16}
Integer Read-Write

The value to assign to ipRouteMetric2, when adding a RIP learned entry to the IP Routing Table. The route preference value for a RIP-learned entry is 4. Note that for route preferences the smaller the value, the higher the preference.

ppeEgpPreference {ppe 17}
Integer Read-Write

The value to assign to ipRouteMetric2, when adding an EGP learned entry to the IP Routing Table. The route preference value for an EGP-learned entry is 2. Note that for route preferences the smaller the value, the higher the preference.

ppeCpuUtilization {ppe 18}
Integer Read-Only

The current CPU utilization of the Packet Processing Engine. Values include:

- low-cpu (1)
- medium-cpu (2)
- high-cpu (3)

ppeRipRouteDiscards {ppe 19}
Counter Read-Only

Number of times a route learned via RIP was not added to the IP Routing Table due to insufficient Routing Table space.

ppeOspfRouteDiscards {ppe 20}
Counter Read-Only

Number of times a route learned via OSPF was not added to the IP Routing Table due to insufficient Routing Table space (currently always zero).

ppeRouteMemorySize {ppe 21}
Counter Read-Only

Number of bytes originally given to the routing table heap.

ppeRouteMemoryAvail {ppe 22}
Counter Read-Only

Number of bytes currently available in the routing table heap.

ppeRouteMemoryFailures {ppe 23}
Counter Read-Only

Number of unsuccessful attempt to allocate from the routing table heap.

ppePacketMemorySize {ppe 24}
Counter Read-Only

Number of bytes originally given to the packet heap.

ppePacketMemoryAvail {ppe 25}
Counter Read-Only

Number of bytes currently available in the packet heap.

ppePacketMemoryfailures {ppe 26}
Counter Read-Only

Number of unsuccessful attempts to allocate from the packet heap. (This value is always zero for Release 3.0.)

ppeOspfPduMemoryFailures {ppe 27}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for an OSPF PDU. (This value is always zero for Release 3.0.)

ppeOspfPduMemoryAllocs {ppe 28}
Counter Read-Only

Number of times memory was allocated for an OSPF BPDU. (This value is always zero for Release 3.0.)

ppeIcmpPduMemoryFailures {ppe 29}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for an ICMP PDU.

ppeIcmpPduMemoryAllocs {ppe 30}
Counter Read-Only

Number of times memory was allocated for an ICMP PDU.

ppeRipPduMemoryFailures {ppe 31}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for a RIP PDU.

ppeRipPduMemoryAllocs {ppe 32}
Counter Read-Only

Number of times memory was allocated for a RIP PDU.

ppeBootpPduMemoryFailures {ppe 33}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for a BOOTP PDU.

ppeBootpPduMemoryAllocs {ppe 34}
Counter Read-Only

Number of time memory was allocated for a BOOTP PDU.

ppeSnmpPduMemoryFailures {ppe 35}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for an SNMP PDU.

ppeSnmpPduMemoryAllocs {ppe 36}
Counter Read-Only

Number of times memory was allocated for an SNMP PDU.

ppeTftpPduMemoryFailures {ppe 37}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for a TFTP PDU.

ppeTftpPduMemoryAllocs {ppe 38}
Counter Read-Only

Number of times memory was allocated for a TFTP PDU.

ppeTraceroutePduMemoryFailures {ppe 39}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for a TraceRoute PDU.

ppeTraceroutePduMemoryAllocs {ppe 40}
Counter Read-Only

Number of times memory was allocated for a TraceRoute PDU.

ppeArpPduMemoryFailures {ppe 41}
Counter Read-Only

Number of unsuccessful attempts to allocate memory for a ARP PDU.

ppeArpPduMemoryAllocs {ppe 42}
Counter Read-Only

Number of time memory was allocated for a ARP PDU.

ppeIgmpPduMemoryFailures {ppe 43}
Counter Read-Only

Number of unsuccessful attempts to allocated memory for a IGMP PDU.

ppeIgmpPduMemoryAllocs {ppe 44}
Counter Read-Only

Number of times memory was allocated for a IGMP PDU.

ppeAresAsStes {ppe 45}
Boolean Read-Write

If true, all-routes-explorers will be treated as if they were spanning tree explorers. Values include:

- true (1)
- false (2)

ppeRoutePercent {ppe 46}
Integer Read-Only

10.3.3.4 Spanning Tree Group

st {admin 4}

These attributes define the ATX wide parameters for the Spanning Tree Protocol.

stGroupAddr {st 1}
MacAddress Read-Write

The Spanning Tree Group address, normally 01-80-c2-00-00-00. Changing stGroupAddr will not take effect until the unit reboots.

stResAddr {st 2}
OctetString Read-Write

A series of 90 octets containing the Spanning Tree's 15 reserved addresses (01-80-c2-00-00-01 through 01-80-c2-00-00-0f). Changing stResAddr will not take effect until the unit reboots.

stBridgeId {st 3}
Obsolete

stRootMaxAge {st 4}
Obsolete

stRootHello {st 5}
Obsolete

stRootDelay {st 6}
Obsolete

stRootId {st 7}
Obsolete

stRootCost {st 8}
Obsolete

stRootPort {st 9}
Obsolete

stTopChange {st 10}
Boolean Read-Only

Whether a topology change is currently in progress.

stActMaxAge {st 11}
Obsolete

stActHello {st 12}
Obsolete

stActDelay {st 13}
Obsolete

stTopChangeCount {st 14}
Integer Read-Write

Number of spanning tree topology changes which have occurred, since the unit was last booted (or this parameter was last reset by the NMS).

stTopChangeTime {st 15}
TimeTicks Read-Only

If stTopChange is True, then the time since stTopChange changed from False to True. If stTopChange is False, then the time since stTopChange changed from True to False.

stAgeTime {st 16}
Obsolete

10.3.3.5 Mesh Group

mesh {admin 5}

This optional group contains the unit wide parameters which control the Cabletron Mesh algorithm. Since the MESH bit within configFunctions is obsolete, this entire Mesh Group is obsolete.

meshCostPercent {mesh 1}
Obsolete

meshCost {mesh 2}
Obsolete

meshCostChange {mesh 3}
Obsolete

meshCostChangeCount {mesh 4}
Obsolete

meshCostChangeTime {mesh 5}
Obsolete

meshSubnet {mesh 6}
Obsolete

10.3.4 Software Distribution Group

swdis {ecs-1 4}

This attribute group provides the means by which the NMS may monitor and/or control the TFTP requests to download a new set of software to the ATX.

swdisDesc {swdis 1}
Octet String Read-Only

The description (0-32 bytes) of the software set currently being downloaded, or a description of the software set last downloaded since the ATX booted. See swDesc.

swdisAccess {swdis 2}
Integer Read-Write

The version of software that may be downloaded. Attempts to download other versions of software will be rejected. The value for swdisAccess is embedded within the software files that are to be distributed via TFTP. Its value may be derived from the names of the software files. For example, file name dnld_hdr.10.2 indicates that the file contains Version 10.2 software. The corresponding value for swdisAccess would be 523, and can be determined by the following algorithm start with the number 1, add the first number, add 256 times the second number, add 256 times 256 times the third number (if present), and add 256 times 256 times 256 times the fourth number (if present). In addition, the NMS operator may wish to use the following special values:

- protected (1) - No software downloads will be accepted
- any-software (2) - A software download of any version of new software will be accepted - this is the factory default

swdisWriteStatus {swdis 3}
Integer Read-Only

The status of the erase/write operation. The possible values are:

- in-progress (1) - An operation is currently in progress
- success (2) - The last operation completed successfully
- config-error (3) - Configuration EPROM encountered an error
- flash-error (4) - Flash EPROM encountered an error
- config-and-flash-errors (5) - Both Configuration EPROM and Flash EPROM encountered errors

swdisConfigIp {swdis 4}
IP Address Read-Write

The IP address of the NMS (i.e., any IP host) that is allowed to use TFTP to send/retrieve the configuration file of the ATX. A value of 0.0.0.0 (which is the factory specified default) prevents any NMS from obtaining access, while a value of 255.255.255.255 gives every NMS permission.

swdisConfigRetryTime {swdis 5}
Integer Read-Write

The number of seconds that the ATX will wait for an acknowledgment before it retransmits an unacknowledged TFTP data block. The factory specified default is 5 seconds.

swdisConfigTotalTimeout {swdis 6}
Integer Read-Write

The number of seconds that the ATX will wait for an acknowledgment to a data block before it cancels a TFTP session. The factory specified default is 25 seconds.

10.3.5 Addresses Configuration Group

addr {ecs-1 5}

This attribute group allows the NMS to examine the current address table entries, as well as to remove both static and learned entry information, or to add and modify pre-defined entry information (modifying certain entry information may cause the entire entry to be deleted), or to zero an entry's statistics.

The Addresses Configuration Group contains a few statistics about the entire Filtering Database, and information about the entry last examined or modified by the NMS. Therefore, the SNMP GetNextRequest PDU is not appropriate to read the entire Filtering Database. Instead, the NMS must perform the following steps:

Step 1. Set addrIndex with the desired starting point (-1 will cause the database to be read from the beginning).

Step 2. Set addrIndexMesh to indicate whether any type of entry is desired, or just Cabletron Mesh Algorithm bridge addresses. The addrIndexMesh parameter does not have to be reset between successive read-next operations.

Step 3. Set addrOperation with the "read-next" operation code. This will cause the ATX to update the Addresses Configuration Group fields about the entry, which includes updating addrNext to contain the value for addrIndex if one desires to obtain the next entry (addrNext will be set to -1 if there is no next entry).

Step 4. Set the desired fields to NULL within the SetRequest PDU, or, issue a GetRequest(s) for the desired fields.

If the addrFlags field is returned with a value of zero, then no entry has been returned, and the NMS should continue trying to read-next, until addrNext is returned with a value of -1.

To read a single specific entry, the NMS must perform the following steps:

Step 1. Set addrMAC with the desired MAC address.

Step 2. Set addrOperation with the "read-random" operation code. This will cause the ATX to update the Addresses Configuration Group fields about the entry.

Step 3. Set the desired fields to NULL within the SetRequest PDU, or, issue a GetRequest (s) for the desired fields.

If the addrFlags field is returned with a value of zero, then the desired entry does not exist.

To remove an entry completely, the NMS must perform the following steps:

Step 1. Set addrMAC with the desired MAC address. (Certain addresses will not be removed by the ATX, i.e., those addresses with entry-other already set within addrFlags.)

Step 2. Set addrOperation with the “delete” operation code. Although the ATX will remove the entry, it may relearn and recreate the entry.

To define a new address or to modify the pre-defined information of an existing address, the NMS must perform the following steps (the order of the steps is not important, as long as addrOperation is set last).

Step 1. Set addrMAC with the desired MAC address. (Certain addresses will not be modified by the ATX, i.e., those addresses with entry-other already set within addrFlags.)

Step 2. Set addrFlags with either entry-static or entry-none, or entry-mesh. (For entry-static and entry-none, either or both of entry-no-source and entry-no-multi may be set. Use of entry-mesh is reserved for the future.) The entry will be deleted (if there is no learned information) when the ATX is told to set addrOperation to “update”, unless any of the following is true: entry-static is set, entry-no-source is set, entry-no-multi is set.

Step 3. If the entry is entry-static, then set addrPortMap and addrBridgeMap. (Use of addrBridgeMap is reserved for the future.)

If the entry is entry-mesh, then additional fields might be added in the future.

Step 4. Set addrOperation with the “update” operation code.

To zero the packet count and character count statistics of an existing address, the NMS must perform the following steps:

Step 1. Set addrMAC with the desired MAC address.

Step 2. Set addrOperation with the “zero-starts” operation code. This will cause the ATX to update the Addresses Configuration Group fields about the entry, before zeroing the statistics.

Step 3. Optionally, set any desired fields to NULL within the SetRequest PDU, or, issue a GetRequest (s) for the desired fields.

If the NMS wants to create a map of which devices exist on which LANs, then the NMS should read all of the dynamically learned addresses from all bridges, and use the spanning tree states of the bridges' ports to determine the devices actual locations.

To read all of a ATX's dynamically learned addresses, the NMS must perform the following steps:

Step 1. Set addrIndex with the desired starting point (-1 will cause the database to be read from the beginning).

Step 2. Set addrOperation with the "read-block" operation code. This will cause the ATX to update the addrBlockSize and addrBlock parameters about a block of addresses, and update addrNext to contain the value for addrIndex if one desires to obtain the next block of entries (addrNext will be set to -1 if there is no next block).

Step 3. Set the desired fields (typically addrNext, addrBlockSize, and addrBlock) to NULL within the SetRequest PDU, or, issue a GetRequest (s) for the desired fields.

Many of the Addresses Configuration Group parameters have an access code of Write-Ignore, which is like Read-Only since the parameters' values are not changed by SetRequests; however, it is different in that SetRequests do not cause access violations.

The Addresses Configuration Group consists of the parameters described below.

addrStatics {addr 1}
Counter Read-Only

The number of static addresses which are currently stored in the ATX.

addrDynamics {addr 2}
Counter Read-Only

The number of learned addresses in the ATX's address table.

addrDynamicMax {addr 3}
Gauge Read-Write

The maximum number of spanning tree addresses which have been learned since the ATX was last booted (or this parameter was last reset by the NMS).

addrMeshs {addr 4}
Counter Read-Only

The number of ATM, Frame Relay, and/or SMDS addresses in the address table. These addresses are not reflected within addrStatics nor addrDynamics. [ATM, Frame Relay, and SMDS are not yet implemented.]

addrDynamicOverflows {addr 5}
Counter Read-Write

The number times an address was not learned due to insufficient address table space.

addrMeshOverflows {addr 6}
Counter Read-Write

The number of times an ATM, Frame Relay, or SMDS address could not be learned due to insufficient address table space. [ATM, Frame Relay, and SMDS are not yet implemented.]

addrFlags {addr 7}
Integer Read-Write

Flags to describe the use and control of this address entry. Each bit has a different meaning. The NMS must always set either entry-static, entry-none or entry-mesh; however, for GetRequests, the NMS will see either no bits set (if the entry does not exist), or exactly one of the following bits set:

- entry-dynamic-local = 30, if set then this is a dynamically learned customer address, which exists on one of this ATX's ports
- entry-dynamic-remote = 29, if set then this is a dynamically learned customer address, which can be reached via additional ATM, Frame Relay, or SMDS addressing (a future enhancement)
- entry-static = 28, if set then this is a pre-defined customer address, e.g., an IEEE Spanning Tree static address
- entry-other = 27, if set then this is a pre-defined internal address, such as the MAC address of one of the ATX's ports, and exactly one of the special entry bits (defined below) must be set
- entry-none = 26, if set then this is a pre-defined filter for a customer address (if one or more of the restriction bits, defined below, is set), or this address does not exist
- entry-mesh = 25, if set then the ATM, Frame Relay, or SMDS data is valid (that data has not yet been defined).

- entry-statistics = 24, if set then this is a dynamically learned customer address, which exists on one of the units ports; however, the address can not be used for packet forwarding decision making (e.g., the address was learned as part of a packet's source routing information, but Spanning Tree learning was not enabled for the port).
- bit 31 is reserved

Any combination of the restriction bits (bits 23-21) may be set for any type of entry, except entry-other or entry-mesh:

- entry-no-source = 23, if set then do not forward any packets from this source address
- entry-no-multi = 22, if set then do not forward any multicast packets from this source address
- Bit 21 is reserved
- Bits 20-18 are reserved

Exactly one of the special entry bits (bits 17-10) must be set for entry-other entries, and none of the bits may be set for other than entry-other entries:

- entry-all-lmas = 17, if set then this is an all LMAs (Local Management Agent) address, i.e., packets are sent to the LMA and forwarded throughout the bridged network, only if the receiving port is in the proper state. The IEEE Spanning Tree All LANs Bridge Management Group Address should be the only instance of this entry.
- entry-any-lma = 16, if set then this is an address for packets to be sent to the LMA, regardless of the receiving port's state. This is reserved for future enhancements.
- entry-lma = 15, reserved for future enhancements.
- entry-port = 14, if set then this is an address for packets to be sent to the LMA. If the receiving port is the port specified by the addrPort field (defined below), then the packets should be sent to the LMA, regardless of the receiving port's state; otherwise, the packets should only be sent if the receiving port is in the proper state and the port specified by addrPort is in the proper state. Each port's MAC address is an instance of this entry.
- entry-bpdu = 13, if set then this is a bridging protocol address, i.e., an address for packets to be sent to the PPE; there are two such addresses the Bridge Group Address of IEEE 802.1 Part D Spanning Tree Protocol (normally 01-80-C2-00-00-10), and the IBM Source Routing Spanning Tree Explorer Group Address (normally 03-00-00-00-80-00).

- entry-reserved = 12, if set then this is a reserved address, and the packets must be discarded.
- Bits 11-10 are reserved for future expansion.
- Bits 9-0 are reserved.

addrMAC {addr 8}
MAC Address Read-Write

The MAC address for this address definition.

addrPort {addr 9}
Integer Write-Ignore

The port through which this address is connected to the ATX (only valid for entry-dynamic-local, entry-statistics, and entry-port entries).

addrPortMap {addr 10}
Octet String Read-Write

A bitmap of up to 32 sets of 32 bits each (128 octets), which defines which source ports are allowed to forward packets to which destination ports (only valid for entry-static entries). Within each set of four octets, each bit is set if a packet with a matching destination address may be forwarded to that port. For example, bits 21-2 of the 9th, 10th, 11th, and 12th, octets correspond to ports 21-2 for packets received by port 2. Since only the ATX's ports 2 through 21 are available for bridging, all unused bits (i.e., bits 0,1,and 22 through 31 of every octet quadruple), as well as the first 8 octets (i.e., the first two octet quadruples), as well as the last 40 octets (i.e., the octet quadruples for ports 22 through 31), are meaningless.

The ATX can store the data most efficiently when the bits within the octet quadruples are set identically for ports 2 through 21.

addrBridgeMap {addr 11}
Octet String Read-Write

Reserved for ATM, Frame Relay, and SMDS.

addrLeastCount {addr 12}
Obsolete

addrTotalCount {addr 13}
Obsolete

addrCost1 {addr 14}
Obsolete

addrCost2 {addr 15}
Obsolete

addrCost3 {addr 16}
Obsolete

addrCost4 {addr 17}
Obsolete

addrSource1 {addr 18}
Obsolete

addrSource2 {addr 19}
Obsolete

addrSource3 {addr 20}
Obsolete

addrSource4 {addr 21}
Obsolete

addrDest1 {addr 22}
Obsolete

addrDest2 {addr 23}
Obsolete

addrDest3 {addr 24}
Obsolete

addrDest4 {addr 25}
Obsolete

addrStabile {addr 26}
Obsolete

addrTxSource {addr 27}
Obsolete

addrTxDestination {addr 28}
Obsolete

addrOldLeast {addr 29}
Obsolete

addrOldTotal {addr 30}
Obsolete

addrOldSource1 {addr 31}
Obsolete

addrOldSource2 {addr 32}
Obsolete

addrOldSource3 {addr 33}
Obsolete

addrOldSource4 {addr 34}
Obsolete

addrOldDestination1 {addr 35}
Obsolete

addrOldDestination2 {addr 36}
Obsolete

addrOldDestination3 {addr 37}
Obsolete

addrOldDestination4 {addr 38}
Obsolete

addrOldTxSource {addr 39}
Obsolete

addrOldTxDestination {addr 40}
Obsolete

addrOperation {addr 41}
Integer Read-Write

The operation to be performed upon the described address. The possibilities include:

- read-random (1)
- read-next (2)
- zero-stats (3)
- update (4)
- delete (5)
- read-block (6)

addrIndex {addr 42}
Integer Read-Write

The index number to be used for read-next and read-block operations. When the ATX boots, addrIndex is initialized to -1.

addrIndexMesh {addr 43}
Boolean Read-Write

Reserved for when ATM, Frame Relay, or SMDS is implemented, this will indicate whether any type of entry (false) should be returned for read-next operation, or just ATM, Frame Relay, or SMDS addresses (true).

addrNext {addr 44}
Integer Write-Ignore

The next value to use for addrIndex, when you use read-next or read-block operations to read the address table. When the ATX boots, addrNext is initialized to -1.

addrAge {addr 45}
TimeTicks Write-Ignore

The time, in centiseconds, since a packet was last received from the address on a port that was in Spanning Tree learning or forwarding state. If no packets have been received, then addrAge contains the time that the address was created on the ATX.

addrRxPkts {addr 46}
Counter Write-Ignore

The number of packets received from the address on a port that was in Spanning Tree learning or forwarding state (can be zeroed by zero-stats).

addrRxChars {addr 47}
Counter Write-Ignore

The number of characters in the packets received from the address on a port that was in Spanning Tree Learning or Forwarding state (can be zeroed by zero-stats).

addrRxMultiPkts {addr 48}
Counter Write-Ignore

The number of multicast packets received from the address on a port that was in Spanning Tree Learning or Forwarding state (can be zeroed by zero-stats).

addrRxFwdPkts {addr 49}
Counter Write-Ignore

The number of packets received from the address which were forwarded (can be zeroed by zero-stats). This includes packets that were MAC-layer bridged because they were received on a port that was in Spanning Tree forwarding state, and packets that were routed regardless of the port's Spanning Tree state.

addrTxPkts {addr 50}
Counter Write-Ignore

The number of packets transmitted to the address (can be zeroed by zero-stats).

addrTxChars {addr 51}
Counter Write-Ignore[2]

The number of characters in the packets transmitted to the address (can be zeroed by zero-stats).

addrBlockSize {addr 52}
Integer Write-Ignore

The number of addresses contained by addrBlock, defined below. This parameter is only used for dumping blocks of MAC addresses.

---

2. These statistics are only updated if the ATX has been configured to gather extended statistics (ppeExtendStats).

addrBlock {addr 53}
Octet String Write-Ignore

A series of 0-700 octets, which represents 0-100 addresses. Each address consists of a 6 octet MAC Address followed by a 1 byte port number. If the port number is zero, then the address is that of a specially configured address; otherwise, the address is a dynamically learned address. This parameter is only used for dumping blocks of MAC addresses.

addrAlarmMAC {addr 54}
MAC Address Read-Only

The MAC address for an address entry. This parameter is only used by the unit's enterprise specific trap mechanism.

addrAlarmPort {addr 55}
Integer Write-Ignore

The repeater port through which this address is connected to the port indicated by addrPort. This parameter is only valid if the port's srepeaterExtendedStats is enabled, and if the unit's ppeExtendStats is true.

10.3.6 SNMP/SMT Support Group

snmpsmt {ecs-1 6}

This parameter group supplies the NMS with a way to access FDDI devices that support SMT services, but not SNMP services. The NMS requests a unit (which must be on the FDDI ring) to perform an SMT query/response exchange with the desired FDDI device.

To build an FDDI ringmap, the NMS should read the upstream neighbor parameter from a unit. Then the NMS should ask the unit to determine the upstream neighbor of FDDI device "x", where "x" is the unit's upstream neighbor. Next, the NMS should ask the unit to determine the upstream neighbor of FDDI device "y", where "y" is the upstream neighbor of device "x". The NMS should continue this process, until a full circle has been determined, i.e., the determined upstream neighbor is the MAC address of the original unit.

snmpsmtUpstreamReq {snmpsmt 1}
Obsolete

snmpsmtUpstreamRsp {snmpsmt 2}
Obsolete

snmpsmtUpstreamDescriptor {snmpsmt 3}
Obsolete

snmpsmtUpstreamState {snmpsmt 4}
Obsolete

fddismtTable {snmpsmt 5}
Not-Accessible

A list of snmpsmt entries, one per FDDI board.

fddismtEntry {fddismtTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

fddismtIndex {fddismtEntry 1}
Integer Read-Only

A unique value for each SMT. It represents the sequence number of the FDDI board in the unit.

fddismtUpstreamReq {fddismtEntry 2}
MacAddress Read-Write

When this variable is written, the unit converts it into a big-endian FDDI address and issues an SMT NIF request to determine that address's upstream neighbor. When the NIF response is received, snmpsmtUpstreamRsp will hold the address of the upstream neighbor and a trap will be sent to the NMS.

fddismtUpstreamRsp {fddismtEntry 3}
MacAddress Read-Only

The address of the upstream neighbor of the unit whose address was written into snmpsmtUpstreamReq. If the address has not yet been learned, then an address of all zeroes is returned.

fddismtUpstreamDescriptor {fddismtEntry 4}
Octet String Read-Only

A four byte string containing the upstream neighbor's Station Descriptor parameter, as defined by the SMT frame protocols. If the upstream neighbor has not yet been learned then a string of all zeroes is returned.

fddismtUpstreamState {fddismtEntry 5}
Octet String Read-Only

A four byte string containing the upstream neighbor's Station State parameter, as defined by the SMT frame protocols. If the upstream neighbor has not yet been learned (i.e., snmpsmtUpstreamRsp contains all zeroes), then a string of all zeroes is returned.

10.3.7 Cabletron Interfaces Group

sinterfaces {ecs-1 7}

This group consists of Cabletron specific information for the interfaces which are not defined in the standard MIBs.

sifUX {sinterfaces 1}
Integer Read-Only

The maximum number of receive data buffers supported by the unit. The sum of sifSmIRxCnt and sifLrgRxCnt for all ports, plus ppeSmlRxCnt and ppeLrgRxCnt must be less than or equal to sifUX.

sifTable {sinterfaces 2}
Not-Accessible

A list of interface entries; one per port.

sifEntry {sifTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

sifIndex {sifEntry 1}
Integer Read-Only

The port number, beginning with 1 for the first port.

sifSmlRxCnt {sifEntry 2}
Integer Read-Only

The number of small data buffers for receiving packets.

sifLrgRxCnt {sifEntry 3}
Integer Read-Write

The number of large data buffers for receiving packets; must be zero for CSMA and UART ports.

sifUxTxCnt {sifEntry 4}
Integer Read-Write

The maximum size of the port's transmit queue, i.e., the number of packets that can be in the port's transmit queue, waiting to be transmitted.

sifThreshold {sifEntry 5}
Integer Read-Write

Maximum number of combined receive and transmit packet hardware errors before an alarm should be generated. See ifInErrors.

sifThresholdTime {sifEntry 6}
Integer Read-Write

The time period (in seconds) to which sifThreshold applies. A value of zero will disable the sifThreshold alarm.

sifRxQueueThresh {sifEntry 7}
Integer Read-Write

Maximum number of receive packet queue overflow errors before an alarm should be generated.

sifRxQueueThreshTime {sifEntry 8}
Integer Read-Write

The time period (in seconds) to which sifRxQueueThresh applies. A value of zero will disable the sifRxQueueThresh alarm.

sifTxStormCnt {sifEntry 9}
Integer Read-Write

The maximum number of multicasts to transmit with a certain period of time (not applicable for UART ports).

sifTxStormTime {sifEntry 10}
TimeTicks Read-Write

The period of time, in centiseconds, which qualifies sifTxStormCnt (not applicable for UART ports).

sifFilterFlags {sifEntry 11}
Integer Read-Write

Special filter flags. The value can be one of the following:

0x0000	DEFAULT Turns the filter off
0x0001	NO_SOURCE_SAME, broadcast all packets from this port that have identical source and destination addresses.
0x0002	NO_SOURCE_LEARNT, do not forward any packets from this port, except those with a pre-defined source address.
0x0004	NO_LEARNING, do not perform the spanning tree learning process, for the packets received from this port.
0x0008	NO_DEST_LEARNT, do not forward any packets to this port, except those with a pre-defined destination address.

sifCongestTime {sifEntry 12}
TimeTicks Read-Write

When a port becomes congested, the PPE will discard any packets destined to the port for this many centiseconds following the congestion alarm. Possible values are 0 through 5 minutes, except for port 1 (the out-of-band management port) which is always 0.

sifQueueTime {sifEntry 13}
TimeTicks Read-Write

The time, in centiseconds, after which an element in a transmit queue will be flushed.

sifPortCost {sifEntry 14}
Obsolete

sifStPriority {sifEntry 15}
Obsolete

sifFunctions {sifEntry 16}
Integer Read-Only

The current functional state (protocols which have been activated and are operational) of the port. The following values or combinations thereof are supported:

0x0001	NMS - shows if the port is enabled for communicating with the NMS (see ifOperStatus)

0x0002	TBRIDGE - shows if the port is enabled for Transparent IEEE 802.1d bridging (see sprotoBridge)
0x0004	SBRIDGE - shows if the port is enabled for Source Routing MAC bridging (see sprotoBridge)
0x0006	SRTBRIDGE - shows if the port is enabled for Source Routing Transparent bridging (see sprotoBridge)
0x0008	MESH - reserved (always not set)
0x0010	IPROUTE - shows if the port is enabled for IP Routing (see sprotoIpRoute)
0x0020	NO_BPDU - shows if BPDUs are suppressed; if TBRIDGE is set, this disables the sending of BPDUs (see sprotoSuppressBpdus) and keeps the port in forwarding state (this is to provide interoperability with non-compliant IEEE 802.1d implementations, however, if the physical topology has any loops, then LAN segments will most likely be flooded with duplicates of packets).
0x0040	RIP - shows if RIP is enabled on the port for IP Routing (see sprotoRip)
0x0080	SRSPAN - shows if auto-span is enabled on the port for SR or SRT bridging (see dot1dSrPortSTESpanMode)
0x0100	PROXY - shows if Proxy ARP is enabled on the port for IP Routing (see sprotoArpProxy)
0x0400	IPX routing - shows if the port is enabled for IPX routing (see sprotoIpxRoute)
0x0800	ARP - shows if ARP is enabled on the port for IP Routing
0x4000	BOOTP - shows if the port is enabled to relay BOOTP and DHCP packets (see sprotoBootpRelay)
0x8000	FRAME_SWITCH - shows if frame switching is enabled (see sifFrameSwitching)
0x10000	TRUNKING - shows if port is enabled (see sprotoTrunking)
0x20000	IPM (multicast) - shows if the port is enabled for IP multicast routing (see sprotoIpMulticast)
0x40000	APPLETALK - shows if the port is enabled for AppleTalk routing (see sprotoAppleRoute)

These values can be configured through the sprotoTable.

Combining TBRIDGE and SBRIDGE always produces SRTBRIDGE.

sifCongested {sifEntry 17}
Boolean Read-Only

Whether the port is out-band congested.

sifState {sifEntry 18}
Obsolete

sifDesigCost {sifEntry 19}
Obsolete

sifDesigRoot {sifEntry 20}
Obsolete

sifDesigBridge {sifEntry 21}
Obsolete

sifDesigPort {sifEntry 22}
Obsolete

sifRxPackets {sifEntry 23}
Octet String Read-Only

A 100 byte array, containing the number of received packets broken down into 25 categories. The number of packets in each category is encoded as a series of 4 bytes, which should be converted into a 32 bit counter. The 25 categories are determined by combining each possible DEST_xxx value with each possible OUTCOME_xxx value, as listed below:

- DEST_UBRIDGE = 0, a bridge management unicast address
- DEST_MBRIDGE = 1, a bridge management multicast address
- DEST_KNOWN = 2, a known non-management unicast address
- DEST_UNKNOWN = 3, an unknown non-management address
- DEST_MULTI = 4, an unknown non-management multicast address
- OUTCOME_FWD = 0, packet forwarded

- OUTCOME_LOCAL =5, packet filtered (local traffic, or the destination was the ATX itself) - note that OUTCOME_LOCAL + DEST_UNKNOWN (or + DEST_MULTI for packets received on ports running the PPP protocol) indicates that the packet has been temporarily queued pending the results of ARP or IP options processing, and the unit may eventually forward the packet.
- OUTCOME_BLOCK = 10, packet filtered (port blocked from bridging and/or routing)
- OUTCOME_SOURCE = 15, packet filtered (restriction on source port, or received packet's contents prevent it from being forwarded); note that a packet with invalid contents may cause DEST_MULTI/DEST_KNOWN to be set for an intended DEST_MBRIDGE/DEST_UBRIDGE type of packet
- OUTCOME_DEST = 20, packet filtered (destination port restriction)

sifRxChar0s {sifEntry 24}
Counter Read-Only

Number of characters in the forwarded received packets.

sifRxChar1s {sifEntry 25}
Counter Read-Only

Number of characters in the filtered received packets.

sifRxSizeErrors {sifEntry 26}
Counter Read-Only

Number of received packets discarded due to size errors.

sifRxHwFCSs {sifEntry 27}
Counter Read-Only

Number of received packets discarded due to FCS errors.

sifRxQueues {sifEntry 28}
Counter Read-Only

Number of received packets lost because of insufficient receive buffers. (For certain interfaces with multipacket FIFO hardware, this count might be approximate.)

sifRxFFcnts {sifEntry 29}
Obsolete

sifTxPackets {sifEntry 30}
Octet String Read-Only

A 20 byte array, containing the number of transmitted packets broken down into 5 categories. The number of packets in each category is encoded as a series of 4 bytes, which should be converted into a 32 bit counter. The 5 categories are the DEST_xxx values, as previously described.

sifTxCongests {sifEntry 31}
Obsolete - see ifOutDiscards.

sifTxStorms {sifEntry 32}
Counter Read-Only

Number of packets not sent due to protection against a multicast storm.

sifTxDests {sifEntry 33}
Counter Read-Only

Number of packets not sent due to a destination port filtering restriction.

sifErrorsFlag {sifEntry 34}
Boolean Read-Only

Whether the port has exceeded the limits specified by sifThreshold and sifThresholdTime.

sifTxStormFlag {sifEntry 35}
Boolean Read-Only

Whether PPE has invoked multicast storm protection.

sifTxSizes {sifEntry 36}
Counter Read-Only

Number of packets not sent due to size limitations.

sifTxAddr {sifEntry 37}
Octet String Read-Only

A series of 16 MAC addresses (96 octets in total), of the source addresses of the last 16 packets that were not sent due to packet size limitations. If fewer than 16 packets were not sent, then the unspecified MAC addresses will be all zeroes.

sifLan {sifEntry 38}
Integer Read-Write

Up to four group identifiers assigned to the port. This variable is used to specify port groups; the first port group entry is put into the highest order byte and all other entries follow from high order to low order. The sifLan identifier is only used by combination filter testing, i.e., for comparisons against filterSrcLan. Possible values are 0 through 255, for each of the parameter's four bytes.

sifStatisticsTime {sifEntry 39}
TimeTicks Read-Only

Length of time during which statistics were collected. In particular, the following statistics may be examined to determine the exact utilization rate of a port:

- ifInOctets
- ifInUcastPkts
- ifInNUcastPkts
- ifOutOctets
- ifOutUcastPkts
- ifOutNUcastPkts
- sifRxChars0s
- sifRxChars1s

sifIpAddress {sifEntry 40}
IP Address Read-Write

The IP address assigned to the port. If this is zero, then the port must learn its IP address via Reverse ARP or PPP's IPCP.

sifIpGroupAddress {sifEntry 41}
IP Address Read-Write

Zero, or an IP group address assigned to the port.

sifMaxPacketSize {sifEntry 42}
Integer Read-Write

The Maximum packet size, in bytes (octets), which may be transmitted through this port. This parameter may not be changed, except for PPP ports, in which case, changing sifMaxPacketSize automatically causes ifMtu and pppLinkLocalMRU to change.

sifExpectSqe {sifEntry 43}
Boolean Read-Write

Whether the port should expect SQE, also known as “heartbeat” (not applicable for ports other than CSMA/CD ports).

sifFilterLocal {sifEntry 44}
Boolean Read-Write

Whether the port should attempt to filter local traffic, bypassing all statistics accumulation, rather than passing the local traffic to the Packet Processing Engine. Filtering local traffic is not available for UART and WAN ports.

sifInQLen {sifEntry 45}
Gauge Read-Write

The maximum length ever obtained by the port’s inbound packet queue (in packets). The size of a port’s inbound packet queue is periodically polled (rather than recalculated whenever a packet is added), so sifInQLen is an approximation of the maximum length.

sifFrameSwitching {sifEntry 46}
Integer Read-Write

Valid for Token Ring ports. Options include:

- enabled (1)
- disabled (2)

Default is disabled (2). If enabled, frames may be bridged between token ring ports on the same board at higher throughput and lower latency than is otherwise possible. Address statistics will not reflect any frames forwarded in this manner. This applies only to transparent bridging, and must be enabled on both the entry and the exit port.

sifRingDrops {sifEntry 47}
Counter Read-Only

Only valid for 802.5 Token Ring, Number of times the ring left the open state.

sifAdapterChecks {sifEntry 48}
Counter Read-Only

Only valid for 802.5 Token Ring, Number of times the MAC chip experienced an internal hardware failure and had to be reset.

10.3.8 Cabletron FDDI Interfaces Group

sfddi {ecs-1 8}

This group contains the Cabletron specific characteristics for FDDI interfaces.

sfddiTable {sfddi 1}
Not-Accessible

A list of interface entries; one per FDDI port.

sfddiEntry {sfddiTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

sfddiIndex {sfddiEntry 1}
Integer Read-Only

The unit's port number of the port.

sfddiRxHwAborts {sfddiEntry 2}
Counter Read-Only

The number of packets that were discarded due to receive abort errors.

sfddiRxParitys {sfddiEntry 3}
Counter Read-Only

The number of packets that were discarded due to parity errors.

sfddiRxShorts {sfddiEntry 4}
Counter Read-Only

The number of packets that were discarded because they were too short.

sfddiDpcErrCnts {sfddiEntry 5}
Counter Read-Only

The number of DPC errors (FIOM MIB only).

sfddiDpcErrValue {sfddiEntry 6}
Integer Read-Only

Detailed status of the last DPC error (FIOM MIB only).

sfddiRbcErrCnts {sfddiEntry 7}
Counter Read-Only

The number of RBC errors (FIOM MIB only).

sfddiRbcErrValue {sfddiEntry 8}
Integer Read-Only

Detailed status of the last RBC error (FIOM MIB only).

sfddiTxAsync {sfddiEntry 9}
Integer Read-Write

Number of bytes to reserve in the DPC Buffer Memory for transmitting asynchronous packets.

sfddiShortAddressing {sfddiEntry 10}
Boolean Read-Only

Whether any packets were received with 16 bit MAC addresses.

sfddiSmtConditions {sfddiEntry 11}
Integer Read-Write

SMT conditions which should be reported to the NMS via SNMP traps. Possible values are any combination of the following bits:

0x00000001	RING, any change in the status of Ring_Op
0x00000002	CLAIM, receipt of My_Claim, Higher_Claim, or Lower_Claim
0x00000004	BEACON, receipt of My_Beacon or Other_Beacon
0x00000008	TVX, expiration of tvx
0x00000010	TRT, expiration of TRT and Late_ct non-zero
0x00000020	PHY-INVALID, receipt of PH_Invalid causing the receiver to enter state R0; this bit indicates that the Link Error Rate Threshold Test has failed, and one or more of the PHYs has been removed from a link connection (the lem_reject_ct field within the FDDI port's status can be used by the NMS to determine which links have failed)
0x00000040	RESET, receipt of MAC_reset causing the receiver to enter state R0
0x00000080	OVERFLOW, overflow of frame_ct, error_ct, or lost_ct
0x00000100	DUP, receipt of a frame with the destination address equal to this station's individual address and the received A indicator set

0x00000200 LOST, recognition that this station's transmitted frame was lost in the course of its circulation around the ring

0x00000400 RESTRICTED, any time the R_Flag goes from reset to set; usage of this bit is reserved for future applications

0x00000800 CONGESTION, any time this station sets Ax but does not set Cx.

0x00001000 TOKEN, receipt of a token if the MAC transmitter is in state T2 (transmitting data) or T3 (issuing token); usage of this bit is reserved for future applications

0x00002000 T4, expiration of TRT in MAC transmitter state T4 (claiming) or T5 (beaconing); usage of this bit is reserved for future applications

sfddiSrfConditions {sfddiEntry 12}
Integer Read-Write

SRF conditions which should be reported to the NMS via SNMP Traps. For those SRF policies which are optional (i.e., SRF_NOT_COPIED, SRF_EB_ERROR, and SRF_TRACE), the setting of a bit also indicates that the policy is to be implemented. Possible values include any combination of the following bits:

0x00000001 SRF-FRAME-ERROR, the MAC frame error rate exceeded the threshold rate.

0x00000002 SRF-PORT-LER, a port's LER Alarm has been triggered.

0x00000004 SRF-NOT-COPIED, the MAC frame not copied rate exceeded the threshold rate.

0x00000008 SRF-DUP-ADDR, a duplicate address condition has been detected.

0x00000010 SRF-EB-ERROR, an elasticity buffer error has occurred.

0x00000020 SRF-CONFIG, obsolete

0x00000040 SRF-CONNECT, an undesirable or illegal connection has been attempted.

0x00000080 SRF-TRACE, obsolete

0x00000100 SRF-NEIGHBOR, a MAC neighbor has changed.

0x00000400 SRF-PEER-WRAP, the station is wrapped with a peer mode connection.

0x00000800 SRF-PATH-CHANGE, the MAC path or a port path has changed.

sfddiSmtConditionsStatus {sfddiEntry 13}
Integer Read-Write

Requested SMT conditions which are currently present (see sfddiSmtConditions). Writing to sfddiSmtConditionsStatus is used to acknowledge the SNMP Traps (refer to chapter 11).

sfddiSrfConditionsStatus {sfddiEntry 14}
Integer Read-Write

Requested SRF conditions which are currently present (see sfddiSrfConditions). Writing to sfddiSrfConditionsStatus is used to acknowledge the SNMP Traps (see chapter 11).

sfddiSrfReportLimit {sfddiEntry 15}
Obsolete

sfddiFrameErrorThreshold {sfddiEntry 16}
Obsolete

sfddiNotCopiedThreshold {sfddiEntry 17}
Integer Read-Write

Value for the fddiMACNotCopiedThreshold variable.

sfddiSBFlag {sfddiEntry 18}
Boolean Read-Only

Whether the ECM's optical bypass is stuck, i.e., the current value of the SMT SB_Flag variable.

sfddiRxEbits {sfddiEntry 19}
Counter Read-Only

The number of packets that were discarded because another station had already set the E bit.

sfddiOBSFuseBad {sfddiEntry 20}
Boolean Read-Only

Whether the fuse to the FDDI port's optical bypass is bad.

sfddiThruB {sfddiEntry 21}
Boolean Read-Write

Whether to connect to the FDDI port B (not applicable for SAS).

sfddiStationDescriptor {sfddiEntry 22}
Octet String Read-Only

A four byte string containing the FDDI port's Station Descriptor parameter, as defined by the SMT frame protocol.

sfddiStationState {sfddiEntry 23}
Octet String Read-Only

A four byte string containing the FDDI ports Station State parameter, as defined by the SMT protocols. The following bits indicate one or a combination of station topology conditions:

x00000100 Peer Wrap - set when the station is in a peer wrap condition (see fddiSMTPeerWrapFlag).

x00000200 Unattached Concentrator - set when a concentrator has no active A, B, or S port.

x00000400 Twisted Ring - set when a A-A connection is detected in the station.

x00000800 Twisted Ring - set when a B-B connection is detected in the station.

x00001000 Rooted Station - set when the station does not have an active A, B, or S port in tree mode.

x00002000 Status Report - set if SRF protocol is enabled (fddiSMTStatRptPolicy).

The following bits indicate one or a combination of duplicate address conditions:

x00000001 My Duplicate - the value of fddiMACDA_Flag for this MAC address.

x00000002 My UNA Duplicate - the value of fddiMACUNDA_Flag for this MAC address.

sfddiDownstreamNbr {sfddiEntry 24}
MacAddress Read-Only

The MAC address of the FDDI port's downstream neighbor.

sfddiSMTBufferSize {sfddiEntry 25}
Integer Read-Write

The amount of packet buffer space to reserve for processing SMT frames. Possible values range from 9000-18000 (FIOM MIB only).

10.3.9 Cabletron UART Interface Group

suart {ecs-1 9}

This group contains the Cabletron specific characteristics for UART interfaces.

suartTable {suart 1}
Not-Accessible

A list of interface entries; one per UART port.

suartEntry {suartTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

suartIndex {suartEntry 1}
Integer Read-Only

The ATX's port number of the port.

suartBaud {suartEntry 2}
Integer Read-Write

This is the desired baud rate. Only 1200 baud through 19,200 baud are supported by the ATX, but the complete set of possibilities is:

- external-clock (1)
- 1200-baud (2)
- 2400-baud (3)
- 4800-baud (4)
- 9600-baud (5)
- 19200-baud (6)
- 38400-baud (7)
- 56-kilobits (8)
- 1.544-megabits (9)
- 2.048-megabits (10)
- 45-megabits (11)

suartModem {suartEntry 3}
Integer Read-Write

The expected modem signals; definitions of the different values are TBD.

suartIpNeighborAddress {suartEntry 4}
Reserved for future use

suartPPPActive {suartEntry 5}
Reserved for future use

suartAlignmentErrors {suartEntry 6}
Counter Read-Only

Number of received packets with frame alignment errors, since the port was last enabled.

suartOverrunErrors {suartEntry 7}
Counter Read-Only

Number of received packets with data overrun errors, since the port was last enabled.

10.3.10 Filter Group

filter {ecs-1 10}

This attribute group provides the mechanism by which the NMS may create, modify, or delete the ATX's combination filters. A combination filter consists of many tests that are simultaneously applied to a packet, to determine if the packet should be filtered. Several combination filters may exist. Each combination filter (i.e., each set of tests) is used sequentially, and the packet is filtered if any of the combination filters determine that the packet should be filtered.

Two sequential series of combination filters may be created for any port. One series is used for packets that are received by the specified port (these are known as entry filters). The second series is used before transmitting packets out the port (these are known as exit filters).

To read a port's combination filters, the NMS must perform the following steps:

Step 1. Issue a GetRequest for filterCurrentCount to determine the total number of combination filters that exist on the entire ATX.

Step 2. Issue a GetRequest for the fields within {filterEntry 1} to read the first combination filter. If it is for the desired port, then retain the information.

Step 3. Continue issuing GetRequests for the fields within the remaining combination filters, i.e., {filterEntry n}, retaining all desired filter information, until all of the filters have been read.

To modify an existing combination filter for a port, the NMS must perform the following steps:

Step 1. Determine which filter is to be modified.

Step 2. Write the new values for the filter (a single SetRequest should be used for all of the fields which are to be changed).

To delete an existing combination filter for a port, the NMS must perform the following steps:

Step 1. Determine which filter is to be deleted.

Step 2. Set the filterDeleteID field to contain the value of the filter's filter ID. Once the ATX receives this SetRequest, the ATX renumbers all of the filters, thereby invalidating the NMS's combination filters information.

Step 3. Optionally, reread all of the combination filters.

To add a new combination filter for a port, the NMS must perform the following steps:

Step 1. Determine the next available filterIndex (filterCurrentCount plus 1). Do not proceed if there are no available filters.

Step 2. Determine the next available filter ID (filterNextID).

Step 3. Set all of the fields for the filterEntry of the next available filterIndex.

Step 4. Determine which filter should precede the new filter, and set filterAddIndex to contain that filter's filter index. (Set filterAddIndex to zero if there are no previous filters.)

Step 5. Set the filterAddID field to contain the next available filter ID. Once the ATX receives this SetRequest, the ATX renumbers all of the filters, thereby invalidating the NMS's combination filters information.

Step 6. Optionally, reread all of the combination filters.

The filter group consists of the following parameters:

filterMaxCount {filter 1}
Integer Read-Only

The maximum number of combination filters that may be defined for the ATX. This is always 100.

filterCurrentCount {filter 2}
Integer Read-Only

The number of combination filters currently defined for the ATX.

filterDeleteID {filter 3}
Integer Read-Write

A filter identifier used for deleting filters. After the NMS sets this field, the ATX deletes the filterEntry that has a matching filter ID field, and resets this field to zero. The ATX then updates filterCurrentCount, filterNextID, and the entire filterTable.

filterNextID {filter 4}
Integer Read-Only

A filter identifier used for adding filters. See filterAddID.

filterAddID {filter 5}
Integer Read-Write

A filter identifier used for adding filters. After the NMS sets this field, the ATX adds the filterEntry indexed by filterAddIndex. The ATX then updates filterCurrentCount, filterNextID, and the entire filterTable. Note that if filterAddID does not match filterNextID, then the ATX will not add the entry, since the SetRequest for filterAddID must have been a duplicate packet.

filterAddIndex {filter 6}
Integer Read-Write

The index within filterTable, after which a filter should be added. If filterAddIndex contains zero, then the filter should be added as the first filterEntry within filterTable.

filterTable {filter 7}
Not-Accessible

A list of combination filters; one per defined filter. If fewer than filterMaxCount filters have been defined, then an additional filterEntry exists, which is used as part of the sequence for defining a new filter.

filterEntry {filterTable 1}
Not-Accessible

A set of objects for a combination filter. The individual components are described below.

filterIndex {filterEntry 1}
Integer Read-Only

This number always matches the instance of the filterEntry. For example, {filterIndex 3} contains the value "3".

filterID {filterEntry 2}
Integer Read-Only

Identifier assigned to the filter, used for obtaining the value for filterDeleteID so that the filter can be deleted.

filterPortNo {filterEntry 3}
Integer Read-Write

The port number related to this filter.

filterComboType {filterEntry 4}
Integer Read-Write

Whether this is an entry filter (0) or an exit filter (1) type.

filterFlags {filterEntry 5}
Integer Read-Write

A set of flag bits which control the operation of the filter. Filters will not prevent packets from being accepted by the ATX as a final destination (e.g., for NMS packets being used to control the ATX). The following bits are used:

- bit 13 used for filterOffset

  if set, the filterOffset field applies to the data beginning from past the Source Routing information, rather than from the start of the translated packet (presence of the Source Routing information is optional).

- bit 12 used for filterOffset

  if set, the filterOffset field applies to the data beginning from past the IP header, rather than from the start of the translated packet. (The packet must be an IP packet with a valid IP header; if the packet is fragmented, then the packet must be the first fragment.)

- bit 11 used for determining if a filter would have been applied

  if set, this filter should not be used for the forwarding/filtering decision making; however, the ATX must determine if a packet would have been filtered if the filter were to have been used (and filterPktCnts must be updated). This is referred to in LCM as a pseudo filter.

- bit 10 used to determine if this filter should be combined

  if set, this filter should be combined, using the or operator, with the next filter of the same filterComboType for the same filterPortNo

- bits 9-8, used for filterFrame

  bit 9, if set filter if match

  bit 8, if set filter if not-match

- bits 7-6, used for filterSource, filterSourceEnd, and filterSourceMask

  bit 7, if set filter if source within range

  bit 6, if set filter if source outside of range

- bits 5-4, used for filterDest, filterDestEnd, and filterDestMask

  bit 5, if set filter if destination within range

  bit 4, if set filter if destination outside of range

- bits 3-2, used for filterSrcLan

  bit 3, if set filter if match

  bit 2, if set filter if not-match

- bits 1-0, used for filterField, filterOffset, and filterMask

  bit 1, if set filter if match

  bit 0, if set filter if not-match

filterFrame {filterEntry 6}
Integer Read-Write

Ethernet protocol type (1501-65535), or one (1) to indicate any 802 frame (i.e., those frames which would be forwarded in 802.3 format, when the destination LAN is CSMA/CD), or two (2) to indicate any Ethernet frame (i.e., those frames which would be forwarded in Ethernet format, when the destination LAN is CSMA/CD). Note that a specific Ethernet protocol type may appear in either Ethernet format frames or encapsulated within 802 format frames.

filterSource {filterEntry 7}
MAC Address Read-Write

Beginning source address.

filterSourceEnd {filterEntry 8}
MAC Address Read-Write

Ending source address.

filterDest {filterEntry 9}
MAC Address Read-Write

Beginning destination address.

filterDestEnd {filterEntry 10}
MAC Address Read-Write

Ending destination address.

filterSourceMask {filterEntry 11}
MAC Address Read-Write

A bitmask to be applied to the packet's source address before checking it against filterSource and filterSourceEnd.

filterDestMask {filterEntry 12}
MAC Address Read-Write

A bitmask to be applied to the packet's destination MAC address before checking it against filterDest and filterDestEnd.

filterSrcLan {filterEntry 13}
Integer Read-Write

Either the port number from which the packet is received, or the value of any one of the group identifiers defined by sifLan1 for that port.

filterOffset {filterEntry 14}
Integer Read-Write

Zero-based offset of an arbitrary field within the packet. Depending upon the value of filterFlags, the offset is relative to the start of the translated packet, beginning with the big endian MAC addresses; or, the offset is relative to the start of the data past the IP header (and, the packet must be an IP packet with a valid IP header); or, the offset is relative to the start of the data past the Source Routing information (presence of Source Routing information is optional); Refer to the packet translation and canonical format figures in Appendix A “Packet Translation Procedure”, to determine the desired offsets for translated packets.

filterField {filterEntry 15}
Octet String Read-Write

An 8 octet value to be compared to the contents of the arbitrary field within the packet.

filterMask {filterEntry 16}
Octet String Read-Write

An 8 octet mask to be applied to the arbitrary field within the packet, before comparing it to filterField.

filterThreshold {filterEntry 17}
Integer Read-Write

The number of filter invocations at which an alarm to the NMS should be generated.

filterThreshTime {filterEntry 18}
Integer Read-Write

The length of time, in seconds, to which filterThreshold applies. A value of zero indicates that no alarms should be sent to the NMS. Values greater than 3600 (i.e., one hour) are not allowed.

filterThreshFlag {filterEntry 19}
Integer Read-Only

Whether usage of this filter has exceeded the limits specified by filterThreshold and filter, and the NMS has not yet acknowledged the alarm. Therefore, whenever the NMS reads this value, it will always be False; however, the NMS may receive alarms (i.e., enterprise specific trap PDUs) with this value set to True.

filterPktCnts {filterEntry 20}
Counter Read-Only

The total number of packets which were discarded due to this filter. Under some circumstances when packets are not forwarded (e.g., local traffic, or packets that are addressed to the ATX itself as the final destination), filterPktsCnts is not necessarily incremented when the filtering criteria agrees with the packet's contents.

filterLastSrc {filterEntry 21}
Octet String Read-Only

The source MAC address of the last packet which was discarded due to this filter. Under some circumstances when packets are not forwarded (see filterPktCnts), filterLastSrc is not necessarily set when the filtering criteria agrees with the packet's contents.

filterFast {filterEntry 22}
Obsolete

10.3.11 Change Reboot Characteristics Command Group

reboot {ecs-1 11}

The set of reboot characteristics includes how to partition the PPE's memory for bridging versus routing, and the module types and port types. The NMS may change any of these characteristics while the unit is running; however, none of them will be applied until the unit reboots. The objects are defined as follows.

rebootBridgingMemory {reboot 1}
Integer Read-Write

The percentage (1-99) of the PPE's memory to be allocated for MAC bridging. The PPE's remaining memory will be used for communicating with the NMS and for routing. See ppeBridgingMemory.

rebootSlotTable {reboot 2}
Not-Accessible

A list of the desired occupants of the slots; one per slot. See hwSlotTable.

rebootEntry {rebootSlotTable 1}
Not-Accessible

A set of objects for each slot. The individual components are described below.

rebootIndex {rebootEntry 1}
Integer Read-Only

The slot number, beginning with 1 for the first slot. This number always matches the instance of the rebootEntry. For example, {rebootIndex 3} contains the value "3".

rebootType {rebootEntry 2}
Integer Read-Write

The desired module type of the slot. See hwDefType.

rebootUseMod {rebootEntry 3}
Integer Read-Write

Whether to use this module, not applicable for the Packet Processing Engine. Possible values 1 or 3. See hwUseMod.

rebootPortType {rebootEntry 4}
Octet string Read-Write

This is a 4 byte array each element represents the types of the ports belonging to this module (see hwPortType).

rebootConfig {reboot 3}
Integer Read-Write

The configuration file that will be used when the unit reboots. This parameter is normally automatically set by the unit; however, an NMS may explicitly set it to either no-change (1) or revert-to-defaults (3). The possible values include:

- no-change (1) - the unit will not change its configuration parameters when it re-boots.
- tftp-config (2) - the unit will use a file that was sent via TFTP when it re-boots.
- revert-to-defaults (3) - the unit will revert to its factory specified defaults when it re-boots.

rebootRouteMemory {reboot 4}
Integer Read-Write

Percentage of non-bridging memory to allocate for routing tables. The remainder will be used for transient packets such as ARP and RIP.

10.3.12 Debug Command Group

debug {ecs-1 12}

This attribute group provides access to the built-in debugging resources. Access to these functions requires the use of the configAnyPass.

Examining the ATX's memory is EXTREMELY DANGEROUS, and may cause it to reboot.

To examine a ATX's memory, the NMS must perform the following steps:

Step 1. Set debugBase with the desired base address.

Step 2. Set debugLength with the desired location count.

Step 3. Set debugData to NULL, so that the ATX's GetResponse PDU will include the current value of debugData (i.e., the contents of the desired memory locations).

Step 4. Set debugOperation with the "DEBUG_EXAMINE" value. All of the above "sets" must be performed in a single SetRequest PDU, with debugOperation being the last parameter. This will cause the ATX to read the desired memory locations, update the debugLength and debugData parameters, and return the updated parameters in the GetResponse PDU.

Modifying an ATX's memory is also EXTREMELY DANGEROUS, and may cause it to reboot.

To modify a ATX's memory, the NMS must perform the following steps:

Step 1. Set debugBase with the desired base address.

Step 2. Set debugLength with the desired location count.

Step 3. Set debugData to contain the desired new data.

Step 4. Set debugOperation with the “DEBUG_MODIFY” value. All of the above “sets” must be performed in a single SetRequest PDU, with debugOperation being the last parameter. This will cause the ATX to modify the desired memory locations, update the debugLength parameter, and return the updated parameter in the GetResponse PDU.

debugStringId {debug 1}
Integer Read-Only

The number of debug text strings sent by the unit. This parameter is used by the ATX’s enterprise specific trap mechanism.

debugString {debug 2}
Octet String Read-Only

The last debug text string (0-80 bytes) sent by the unit.

debugTable {debug 3}
Not-Accessible

A list of the modules which may be debugged, with one module per slot defined by hwSlotTable.

debugEntry {debugTable 1}
Not-Accessible

A set of objects for each module. The individual objects are described below.

debugIndex {debugEntry 1}
Integer Read-Only

The module number, beginning with 1 for the module in the first slot. This number always matches the instance of the debugEntry. For example, {debugIndex 3} contains the value ”3”.

debugOperation {debugEntry 2}
Integer Read-Write

Type of debug operation requested. Possible values include:

- examine (1) – the ATX’s memory should be examined
- modify (2) – the ATX’s memory should be modified

debugBase {debugEntry 3}
Integer Read-Write

The base address of the memory region to be examined or modified.

debugLength {debugEntry 4}
Integer Read-Write

The desired length, in words, of the memory region to be examined or modified. After the ATX performs the operation, debugLength is updated to indicate the length of memory that was actually examined or modified. If debugLength is updated to contain zero, then the module was not operational.

debugData {debugEntry 5}
Octet String Read-Write

A series of 0-64 octets, which represents 0-16 words (this may change in the future, if the ATX has processors with word sizes of other than 4 bytes per word). When this variable is read, the ATX uses the values in debugBase and debugLength to return the contents of a specified memory range to the NMS. Similarly, when this variable is written, a specified area of memory will be overwritten with data supplied by the NMS.

10.3.13 Loopback Command Group

lpbk {ecs-1 13}

The NMS may initiate/terminate local or remote loopback on a port. In local loopback, the port is disconnected from its network. The ATX generates loopback packets for the port, and the port loops back packets without sending them onto its network.

In remote loopback, the port is in normal operation, sending/receiving normal packets. In addition, the ATX generates loopback packets to be sent out the port to a particular destination device on the port's network. That destination device echoes the packet back onto the network, which the port then receives. For ports in remote loopback, the ATX creates LLC Type 1 Test packets for LANs, and PPP Echo Request packets for WANs and UARTs.

To put a port into local loopback, the NMS must perform the following steps

Step 1. Set the lpbkPktLength, lpbkPktNum, lpbkIncrements, lpbkInterval, and lpbkOperation parameters with a single SetRequest PDU, making sure that lpbkOperation is the last parameter. The value loopback-local must be specified for lpbkOperation, and the previous value for lpbkOperation must not be loopback-remote.

Step 2. Wait for the GetResponse PDU. If the port is not disabled (refer to ifOperStatus), then a generic link down trap PDU may be generated.

Step 3. If the GetResponse did not indicate an error, then wait for (and acknowledge) the enterprise specific trap that contains the lpbkOperation parameter. [Multiple traps may be sent if there are multiple errors.]

To put a port into remote loopback, the NMS must perform the following steps

Step 1. Set the lpbkPktLength, lpbkPktNum, lpbkIncrements, lpbkInterval, lpbkDestAddr, and lpbkOperation parameters with a single SetRequest PDU, making sure that lpbkOperation is the last parameter. The value loopback-remote must be specified for lpbkOperation, and the previous value for lpbkOperation must not be loopback-local. Also, the port must be enabled according to ifOperStatus.

Step 2. Wait for the GetResponse PDU.

Step 3. If the GetResponse did not indicate an error, then wait for (and acknowledge) the enterprise specific trap that contains the lpbkOperation parameter. (Multiple traps may be sent if there are multiple errors.)

A port's loopback operation is automatically terminated after the requisite number of packets have been sent. To immediately terminate a port's loopback operation, the NMS may perform the following steps:

Step 1. Issue a SetRequest PDU, specifying the value loopback-off for lpbkOperation.

Step 2. Wait for the GetResponse PDU. If the port was in local loopback, then the port will be automatically re-enabled to the extent specified by the current value of sprotoEntry for that port.

lpbkTable {lpbk 1}
Not-Accessible

A list of interface entries; one per port.

lpbkEntry {lpbkTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

lpbkIndex {lpbkEntry 1}
Integer Read-Only

The port number, beginning with 1 for the first port.

lpbkOperation {lpbkEntry 2}
Integer Read-Write

The type of loopback being performed. Setting lpbkOperation to loopback-off causes loop back operation to terminate. Setting lpbkOperation to loopback-local or loopback-remote causes loopback operation to commence, and the ATX automatically resets lpbkOperation to loopback-off when the loopback operation has completed. Possible values include:

- loopback-off (1) – terminate loopback
- loopback-local (2) – commence local loopback
- loopback-remote (3) – commence remote loopback

lpbkDestAddr {lpbkEntry 3}
MacAddress Read-Write

The MAC address of the target device, only needed for remote loopback.

lpbkPktNum {lpbkEntry 4}
Integer Read-Write

The number of loopback packets to be transmitted.

lpbkInterval {lpbkEntry 5}
TimeTicks Read-Write

The desired time delay between the reception of a looped back packet, and the transmission of the next packet to be looped back.

lpbkPktLength {lpbkEntry 6}
Integer Read-Write

The length (in octets) of the loopback packet. The values of the data itself are automatically selected by the ATX.

lpbkIncrements {lpbkEntry 7}
Integer Read-Write

The number of bytes of data to be added to the loopback packet after each successful loopback packet transmission and reception. When the port's maximum packet size has been reached, the ATX automatically begins again, starting with lpbkPktLength.

lpbkGoods {lpbkEntry 8}
Counter Read-Only

The number of packets successfully looped back. This number is automatically reset to zero, whenever loopback operation is initiated.

lpbkErrorNoReceives {lpbkEntry 9}
Counter Read-Only

The number of packets which were transmitted, but not received. This number is automatically reset to zero, whenever loopback operation is initiated.

lpbkErrorBadReceives {lpbkEntry 10}
Counter Read-Only

The number of transmitted packets which were received with different data patterns. This number is automatically reset to zero, whenever loopback operation is initiated.

lpbkErrorSize {lpbkEntry 11}
Integer Read-Only

The length of the last loopback packet which caused either lpbkErrorNoReceives or lpbkErrorBadReceives to be incremented. This number is automatically reset to zero, whenever loopback operation is initiated.

lpbkErrorSent {lpbkEntry 12}
OctetString Read-Only

The transmitted bytes (0-4) of data within the last loopback packet which caused lpbkErrorBadReceives to be incremented. This parameter is automatically reset to the null string, whenever loopback operation is initiated.

lpbkErrorReceived {lpbkEntry 13}
OctetString Read-Only

The received bytes (0-4) of data within the last loopback packet which caused lpbkErrorBadReceives to be incremented. This parameter is automatically reset to the null string, whenever loopback operation is initiated.

lpbkErrorOffset {lpbkEntry 14}
Integer Read-Only

The byte offset (one being the first byte) within the loopback packet where lpbkErrorSent begins. This number is automatically reset to zero, whenever loopback operation is initiated.

10.3.14 Cabletron WAN Interface Group

swan {ecs-1 14}

This group contains the Cabletron specific characteristics for WAN interfaces.

swanTable {swan 1}
Not-Accessible

A list of interface entries; one per WAN port.

swanEntry {swanTable 1}
Not-Accessible

A set of objects for an interface entry. The individual components are described below.

swanIndex {swanEntry 1}
Integer Read-Only

The unit's port number of the port.

swanDesiredSpeed {swanEntry 2}
Obsolete

swanActual Speed {swanEntry 3}
Integer Read-Only

This is the actual line speed. The possible values are defined by suartSpeed. For HSSI, the speed is 45 megabits (11). If the RS-449 interface is being used on the HSSI IOM, the speed is always reported as 1.544 megabits (9); however, the external clock may be generating a slower speed.

swanIpNeighborAddress {swanEntry 4}
IpAddress Read-Write

Zero, or an IP address to assign to the port's neighbor.

swanPPPActive {swanEntry 5}
Obsolete

swanAlignmentErrors {swanEntry 6}
Obsolete

swanOverrunErrors {swanEntry 7}
Counter Read-Only

Number of received packets with data overrun errors, since the port was last enabled.

swanPortType {swanEntry 8}
Obsolete

swanLinkCost {swanEntry 9}
Obsolete

swanMeshState {swanEntry 10}
Obsolete

swanLinkSubnet {swanEntry 11}
Obsolete

swanLinkBridge {swanEntry 12}
Obsolete

swanLinkPort {swanEntry 13}
Obsolete

swanNegotiate {swanEntry 14}
Obsolete

swanSwitches {swanEntry 15}
Integer Read-Write

Current port selected. Possible values include:

- wan-hssi (1) - HSSI is selected
- wan-t1 (2) - RS-449 is selected
- wan-hssi-switching-disabled (3) - HSSI is selected and cannot be switched to RS-449 except by NMS interaction
- wan-t1-switching-disabled (4) - 449 is selected and cannot be switched to HSSI except by NMS interaction

swanDCEDrops {swanEntry 16}
Counter Read-Only

Number of instances when loss of DCE has been detected.

10.3.15 Cabletron Repeater Interface Group

srptr {ecs-1 15}

This entire group is obsolete.

10.3.16 Cabletron Repeater Interface Group

srepeater {ecs-1 16}

This group contains the Cabletron specific characteristics for CSMA/CD interfaces which have embedded repeater capabilities.

srepeaterTable {srepeater 1}
Not-Accessible

A list of interface entries; one per CSMA/CD port that has embedded repeater capabilities. Enterprise-specific status information and control variables for a collection of repeaters in a particular system.

srepeaterEntry {srepeaterTable 1}
Not-Accessible

Enterprise-specific status information and control variables for a particular repeater in the system.

srepeaterGroupID {srepeaterEntry 1}
Integer Read-Only

The value of srepeaterGroupID is assigned so as to uniquely identify a repeater in a system where a single agent manages multiple repeaters. Values range from 1 to 1024.

srepeaterLinkStatusMask {srepeaterEntry 2}
Integer Read-Only

This mask provides quick access to the general connectivity of the repeater. Each bit corresponds to the link status of a repeater port. If the bit is 1 then the link pulses are being received, as per the IEEE specification. A bit value of zero indicates one of four conditions: 1) There is nothing physically attached to the port; 2) The port is not configured for twisted pair. The AUI interface, for example, does not generate or test for link pulses; 3) The device attached to the port does not generate the link pulses specified by the IEEE standard. STARLAN devices, for example, do not generate such pulses; 4) The port is configured to ignore link pulses (see srepeaterPortLinkPulse). The least significant bit in the mask corresponds to repeater port 1; the high-order eight bits are always zero.

srepeaterExtendedStats {srepeaterEntry 3}
Integer Read-Only

When enabled (1) the rptrMon, rptrMonPort, and rptrAddrTrack tables are available. The unit's packet processing performance may be slightly degraded due to the extra processing required, although the effect will not ordinarily be noticed. When disabled (2), the rptrMon, rptrMonPort, and rptrAddrTrack tables are not available.

srepeaterPortTable {srepeater 2}
Not-Accessible

Enterprise-specific status information and control variables for a collection of repeater ports in a particular system. As with the standard repeater MIB, this table treats a CEIOM, which is actually a complete repeater domain, as a repeater group.

srepeaterPortEntry {srepeaterPortTable 1}
Not-Accessible

Enterprise-Specific status information and control variables for a particular repeater port in the system.

srepeaterPortGroupID {srepeaterPortEntry 1}
Integer Read-Only

The value of srepeaterPortGroupID is assigned so as to uniquely identify a repeater in a system where a single agent manages multiple repeaters. This corresponds to ifIndex, as well as rptr...GroupIndex.

srepeaterPortPortID {srepeaterPortEntry 2}
Integer Read-Only

This variable identifies the port within the repeater (CEIOM) for which this entry provides management information. For the CEIOM with 24 UTP ports, the leftmost position id port 1, and the rightmost is port 24.

srepeaterPortLinkPulse {srepeaterPortEntry 3}
Integer Read-Write

If enabled (1), the repeater port is to generate and test for link pulse. If disabled (2), the repeater port is to neither generate nor test for link pulse. Link pulse is a means of verifying connectivity between the repeater and the end station (see srepeaterLinkStatusMask), as specified by the 10BASE-T standard. The ability to disable link pulse is provided for compatibility with equipment which antedates the 10BASE-T standard.

10.3.17 Cabletron Protocol Group

sproto {ecs-1 17}

This group specifies which protocols apply to each interface and allows configuration of bridging-translation between Token Ring and Ethernet for various protocols.

sprotoTable {sproto 1}
Not-Accessible

This table contains configuration information specifying the types of protocols used to bridge, route, translate, and filter frames for each port.

sprotoIpxDot3Framing {sproto 2}
Integer Read-Write

If sprotoIpxTranslate is enabled, specifies the type of framing to be used for NetWare IPX packets on 802.3 networks:

- ethernet8023 (1) - specifies that the 802.3 header is to be used without an 802.2 header
- ethernet2 (2) - specifies that Ethernet-2 framing is to be used, with ethernet Type 8137
- ethernet8022 (3) - specifies that an LLC header e0,e0,03 is to be used in conjunction with the 802.3 header.

sprotoEntry {sprotoTable 1}
Not-Accessible

Each entry in this table specifies which protocols are used by a particular port in the ATX.

sprotoIfIndex {sprotoEntry 1}
Integer Read-Only

Identifies the interface (port) to which this entry's information belongs. The value of this variable corresponds to ifIndex, as well as to most of the other port identification values in related MIBs.

sprotoBridge {sprotoEntry 2}
Integer Read-Write

This variable defines the bridging method to be applied to frames received from, or destined to, this port. Ethernet and FDdI ports may be configured for:

- transparent (1)
- srt (3)
- none (4)

Token Ring ports may be configured for any of the options listed above and also for:

- sr (2)

The UART may only be configured to none.

sprotoSuppressBpdus {sprotoEntry 3}
Integer Read-Write

Allows transmission of spanning tree protocol packets to be suppressed. If norman (1), 802.1d spanning tree packets and source-routing spanning tree packets are transmitted as usual. If suppressed (2), these packets are not transmitted. This option may be needed for interoperability with non-802.1d spanning tree protocols (e.g., DEC LanBridge 100).

sprotoIpRoute {sprotoEntry 4}
Integer Read-Write

Specifies whether or not this port is to be used for IP routing. either:

- enabled (1)
- disabled (2)

sprotoIpxRoute {sprotoEntry 5}
Integer Read-Write

Specifies whether or not this port is to be used for Novell NetWare or IPX routing. either:

- enabled (1)
- disabled (2)

sprotoAppleRoute {sprotoEntry 6}
Integer Read-Write

Specifies whether or not this port is to be used for AppleTalk routing. either:

- enabled (1)
- disabled (2)

sprotoArpTranslate {sprotoEntry 7}
Integer Read-Write

These options are intended to compensate for any likely implementation of internet ARP on 802.5 Token Ring. Options Include:

- none (1)
- bitswap1 (2) - then embedded MAC addresses for ARP packets with hardware type 1 (Ethernet) will be bitswapped.
- bitswap6 (3) - then embedded MAC addresses for ARP packets with hardware type 6 (IEEE 802) will be bitswapped.
- bitswap61 (4) - both type 1 and type 6 ARP packets will be bitswapped.
- oneto6 (5) - then in-coming type 6 ARP packets will be changed to type 1 and outgoing type 1 will be changed to type 6, without bitswapping.
- oneto6swap (6) - then in-coming type 6 ARP packets will be converted to type 1 and outgoing type 1 ARP packets will be changed to type 6, with embedded addresses bitswapped.

This option only applies to Token Ring ports. The factory default for all ports is none (1). Values other than (1) and (6) are deprecated.

sprotoArpSourceRoute {sprotoEntry 8}
Integer Read-Write

Specifies whether, when bridging IP ARP packets which are source routing explorer frames, the source routing information is to be stripped or propagated. If pass-Rif (1), the source frame is bridged as is, with route-discovery proceeding as expected. If stripRif (2), then the routing information field is stripped before being propagated. This option allows non-source-routing (e.g., Ethernet) IP hosts to communicate transparently. If passBoth (3), then both the original source-routed frame and the transparent equivalent are propagated. This last option provides maximum connectivity at some expense in network traffic.

This option applies only to Token Ring ports.

sprotoIpxTranslate {sprotoEntry 9}
Integer Read-Write

Specifies whether, when bridging Novell IPX frames, they are to be translated to Ethernet-like frame format. This option applies only to Token Ring ports. Either:

- enabled (1)
- disabled (2)

sprotoAppleTranslate {sprotoEntry 10}
Integer Read-Write

Specifies whether, when bridging AppleTalk, ELAP, TLAP, and AARP frames, those frames are to be translated between Token Ring and Ethernet format before being forwarded. This option only applies to Token Ring ports. Either:

- enabled (1)
- disabled (2)

sprotoIbmLlcTranslate {sprotoEntry 11}
Integer Read-Write

Specifies whether, when bridging IBM LLC frames, an 8209 style translation between Token Ring and Ethernet format is to be performed before being forwarded. This option only applies to Token Ring ports. Either:

- enabled (1)
- disabled (2)

sprotoRip {sprotoEntry 12}
Integer Read-Write

Specifies whether the internet Routing Information Protocol is to be used over this port. Only valid if sprotoIpRoute is enabled. Either:

- enabled (1)
- disabled (2)

sprotoEgp {sprotoEntry 13}
Integer Read-Write

Specifies whether the Internet External Gateway Protocol is to be used over this port. Only valid if sprotoIpRoute is enabled (not currently supported). Either:

- enabled (1)
- disabled (2)

sprotoOspf {sprotoEntry 14}
Integer Read-Write

Specifies whether the internet Open Shortest Path First protocol is to be used over this port. Only valid if sprotoIpRoute is enabled (not currently supported). Either:

- enabled (1)
- disabled (2)

sprotoArpProxy {sprotoEntry 15}
Integer Read-Write

Specifies whether the port is to respond to internet ARP requests for which the unit is the next hop in a routed path. Only valid if sprotoIpRoute is enabled. Either:

- enabled (1)
- disabled (2)

sprotoIbm8209Lsaps {sprotoEntry 16}
Octet String Read-Write

Specifies up to eight LSAPs for which IBM 8209 style Token Ring to Ethernet encapsulation is to be performed. Only valid if sprotoIbmLlcTranslate is enabled.

sprotoBootpRelay {sprotoEntry 17}
Integer Read-Write

Specifies whether or not this port is to relay BOOTP packets, as described in RFC-951 and subsequent internet drafts. Only valid if sprotoIpRoute is enabled. Either:

- enabled (1)
- disabled (2)

sprotoNetbiosTranslate {sprotoEntry 18}
Integer Read-Write

Specifies whether, when bridging netbios packets which are source routing explorer frames, the routing information is to be stripped or propagated. Options include:

- passRif (1)
- stripRif (2)
- passBoth (3)

- none (4)

If passRif (1), the frame is bridged as is, with route discovery proceeding as expected. This option allows non-source routing (e.g., Ethernet) netbios hosts to communicate transparently. If stripRif (2), then the routing information field is stripped before being propagated. If passBoth (3), then both the original source-routed frame and the transparent equivalent are propagated. This choice provides maximum connectivity at some expense in network traffic. None (4) is identical to passRif.

This option applies only to Token Ring ports.

sprotoIpMulticast {sprotoEntry 19}
Integer Read-Write

Specifies whether the internet Multicast Routing Protocol is to be used over this port. Only valid if sprotoIpRoute is enabled.

sprotoTrunking {sprotoEntry 20}
Integer Read-Write

Specifies whether Cabletron's trunking protocol (an extension to the standard Spanning Tree Bridging protocol) is to be used over this port.

10.3.18 Cabletron IPX Group

sipx {ecs-1 18}

This group contains the IPX routing information.

10.3.18.1 SIPXIF Table

sipxIfTable {sipx 1}
Not-Accessible

This table contains per-interface attributes of IPX routing.

sipxIfEntry {sipxIfTable 1}
Not-Accessible

Each entry (indexed by sipxIfIndex) defines the IPX routing information used by a particular port in the unit. Entries exist for all interfaces, whether or not they are actually populated or configured for IPX routing.

sipxIfIndex {sipxIfEntry 1}
Integer Read-Only

Identifies the interface (port) to which this entry's information belongs. The value of this variable corresponds to ifIndex, as well as to most of the other port identification values in related MIBs.

sipxIfNetwork {sipxIfEntry 2}
OctetString Read-Write

The IPX network number (four bytes) associated with this interface.

sipxIfFraming {sipxIfEntry 3}
Integer Read-Write

Defines the link-level framing to be used for this interface. Options Include:

- ethernet8023 (1) - framing uses an 802.3 length followed by the IPX header and data (the first 2 bytes of the IPX header are hexadecimal ffff). This is the default for Ethernet links.
- ethernet2 (2) - framing uses an Ethernet type of hexadecimal 8137, followed by the IPX data.
- ieee8022 (3) - framing uses DSAP and LSAP fields of hexadecimal e0 followed by a control byte of 03 (unnumbered information), followed by the IPX data. This is the default for non-Ethernet links (Ethernet links precede the SAPs with an Ethernet length).
- snap (4) - framing uses standard SNAP encapsulation of the 8137 Ethernet type, followed by the IPX data.
- rawfddi (5) - framing uses an FDDI MAC header followed immediately by the IPX header. There are neither LLC nor SNAP headers.
- ppp (6) - the IPX packet is encapsulated over the PPP protocol, for use over a wide-area network.

sipxIfInRipPkts {sipxIfEntry 4}
Counter Read-Only

Number of IPX routing information protocol packets received on this interface.

sipxIfOutRipPkts {sipxIfEntry 5}
Counter Read-Only

Number of IPX routing information protocol packets transmitted on this interface.

sipxIfInSapPkts {sipxIfEntry 6}
Counter Read-Only

Number of IPX Service Advertising Protocol packets received on this interface.

sipxIfOutSapPkts {sipxIfEntry 7}
Counter Read-Only

Number of IPX Service Advertising Protocol packets transmitted on this interface.

10.3.18.2 Route Table

sipxRouteTable {sipx 2}
Not-Accessible

This table contains an entry for each route presently known to this router.

sipxRouteEntry {sipxRouteTable 1}
Not-Accessible

Each entry (indexed by sipxRouteDest) is a route to a particular destination.

sipxRouteDest {sipxRouteEntry 1}
Octet String Read-Only

The destination network address of this route.

sipxRouteIfIndex {sipxRouteEntry 2}
Integer Read-Only

The index value which uniquely identifies the local interface through which the next hop of this route should be reached. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

sipxRouteTickCount {sipxRouteEntry 3}
Integer Read-Only

The primary routing metric for this route, which is an estimate of the number of 55 millisecond ticks that are needed to reach the destination.

sipxRouteHopCount {sipxRouteEntry 4}
Integer Read-Only

The secondary routing metric for this route, which is the number of routers that must be traversed to reach the destination.

sipxRouteNextHop {sipxRouteEntry 5}
Octet String Read-Only

The IPX node address of the next hop of this route, if indirect. If direct, the address of the local interface.

sipxRouteAge {sipxRouteEntry 6}
Counter Read-Only

The number of seconds since this route was last updated.

10.3.18.3 SAP Table

sipxSapTable {sipx 3}
Not-Accessible

This table contains an entry for each SAP-cache service presently known to this router.

sipxSapEntry {sipxSapTable 1}
Not-Accessible

Each entry (indexed by sipxSapIndex) is an IPX service definition, as advertised by the SAP protocol and distributed through the SAP cache of this router.

sipxSapIndex {sipxSapEntry 1}
Integer Read-Only

Used only as an artificial index to this table; has no independent meaning.

sipxSapType {sipxSapEntry 2}
Integer Read-Only

Identifies the type of service provided by the server. Values range 0..ffff, and are defined by Novell. For example, file service is type 4.

sipxSapName {sipxSapEntry 3}
Octet String Read-Only

Server Name (no more than 48 characters).

sipxSapNetwork {sipxSapEntry 4}
Octet String Read-Only

The IPX network address of the server providing the indicated service.

sipxSapNodeId {sipxSapEntry 5}
Octet String Read-Only

The IPX node address of the server. With NetWare 2.x, this corresponds to a physical MAC address, represented here in canonical (little-endian) bit order. With NetWare 3.x, it is typically 000000000001.

sipxSapSocket {sipxSapEntry 6}
Integer Read-Only

The socket number to which service requests should be addressed.

sipxSapHopCount {sipxSapEntry 7}
Integer Read-Only

A count of the number of routers that must be traversed to reach this server.

sipxDiscardedRoutes {sipx 4}
Counter Read-Only

A count of the number of received routes that were lost due to insufficient resources.

sipxDiscardedServices {sipx 5}
Counter Read-Only

A count of the number of received SAP services that were lost due to insufficient resources.

10.3.19 Router Discovery Group

This MIB configures the router discovery feature of the unit. The Router Discovery protocol is defined in RFC 1256.

srtrdisc {ecs-1 19}

srtrdiscTable {srtrdisc 1}
Not-Accessible

srtrdiscEntry {srtrdiscTable 1}
Not-Accessible

srtrdiscIfIndex {srtrdiscEntry 1}
Integer Read-Only

Identifies the interface (port) to which this entry's information belongs. The value of this variable corresponds to ifIndex, as well as most of the other port identification values in related MIBs.

srtrdiscState {srtrdiscEntry 2}
Integer Read-Write

Specifies whether or not the router discovery protocol is to be used for this interface. Defaults to disabled (2). Router discovery is not activated unless IP routing is also enabled (1) for this interface. (See sprotoIpRoute.) Values include:

- enabled (1)
- disabled (2)

srtrdiscAdvertisementAddress {srtrdiscEntry 3}
Integer Read-Write

The IP address to be used for multicast router advertisements for this interface. If multicast (1), the all-systems multicast 224.0.0.1 is used. If broadcast (2), the limited-broadcast 255.255.255.255 is used.

srtrdiscAdvertisementInterval {srtrdiscEntry 4}
Integer Read-Write

The interval at which multicast router advertisements are to be transmitted, in seconds. Allowable values range between 4 and 1800.

srtrdiscLifetime {srtrdiscEntry 5}
Integer Read-Write

The value to be used for the lifetime field of router advertisements. Values range from srtrdiscAdvertisementInterval to 9000.

srtrdiscPreference {srtrdiscEntry 6}
Integer Read-Write

The preferability of this interface's address as a default router, relative to other routers on the same subnet.

10.3.20 IP Multicasting Group

sipm {ecs-1 20}

sipmRouteTable {sipmroute 1}
Not-Accessible

The (conceptual) table listing the origin routes that this router learns from neighboring routers and from initial installations of directly-connected routes. Origin is defined to be the subnetwork that can originate multicast packets.

sipmRouteEntry {sipmRoutetable 1}
Not-Accessible

An entry (conceptual row) representing information on an origin route when IPM is enabled.

sipmRouteOrigin {sipmRouteEntry 1}
IP Address Read-Only

The subnetwork address of the origin.

sipmRouteOriginMask {sipmRouteEntry 2}
IP Address Read-Only

The subnetwork mask of the origin.

sipmRouteGateway {sipmRouteEntry 3}
IP Address Read-Only

The next hop gateway toward the origin. This object has the value 0.0.0.0 for directly-connected routes.

sipmRouteMetric {sipmRouteEntry 4}
Integer Read-Only

The distance to this origin in number of hops.

sipmRouteAge {sipmRouteEntry 5}
TimeTicks Read-Write

The time left for this origin route before aged out.

sipmRouteParents {sipmRouteEntry 6}
Octet String Read-Only

The bitmapped incoming port (s) for which this origin is learned. This is an octet string of length 8. The least significant bit of the eighth octet is port 0. The least significant bit of the seventh octet is port 9 and so on up to a maximum of 64 ports.

10.3.20.1 IP Multicast Neighbor Table

sipmNeighborTable {sipmneighbor 1}
Not-Accessible

The table listing the neighboring routers on each interface port.

sipmNeighborEntry {sipmNeighborTable 1}
Not-Accessible

An entry of all neighbors which are learned from this local interface port.

sipmNeighborIfIndex {sipmNeighborentry 1}
Not-Accessible

The ifIndex value of the interface for which this entry contains information.

sipmNeighbors {sipmNeighborEntry 2}
IP Address Read-Only

The neighbor's IP address which was learned.

sipmNeighborLastHeard {sipmNeighborentry 3}
TimeTicks Read-Only

The elapsed time since the neighbor was last heard.

sipmOutOfMemory {sipmstat 1}
Counter read-Write

The number of times IP Multicast routines try to allocate memory for routes which failed.

10.3.20.2 IP Multicast Stat Table

sipmStatTable {sipmstat 2}
Not-Accessible

The table containing multicast routing statistics for IP datagrams on a per port basis.

sipmStatEntry {sipmStatTable 1}
Not-Accessible

An entry containing the multicast routing statistic for IP datagrams on a particular port.

sipmStatIfIndex {sipmStatEntry 1}
Integer Read-Only

The interface port number on which corresponding IPM statistics gather.

sipmStatBadPackets {sipmStatEntry 2}
Counter Read-Write

The number of bad packets received which may include unknown DVMRP/ IGMP messages, short packets, and checksum errors.

sipmCacheMiss {sipmStatEntry 3}
Counter Read-Write

The number of times IP Multicast packets need to lookup a route at the IP layer for determining forwarding path.

10.3.21 IP Circuit Table

sipckt {ecs-1 21}

sipcktTable {sipckt 1}
Not-Accessible

Defines attributes of IP circuits. A circuit is defined as a binding of a network number to a physical interface.

sipcktEntry {sipcktTable 1}
Not-Accessible

Defines the attributes of a binding of IP address and network to physical interface.

sipcktIndex {sipckEntry 1}
Integer Read-Write

Identifies the physical interface. Identical to ifIndex.

sipcktIpAddress {sipckEntry 2}
IP Address Read-Write

Identifies the Ip network that is bound to this interface.

sipcktState {sipckEntry 3}
Integer Read-Write

The status of this binding.

sipcktNetMask {sipckEntry 4}
IP Address Read-Write

The subnet mask for the network associated with this address binding. If set to 0.0.0.0, will default to standard class A/B/C mask, depending on the value of sipcktIpAddress.

sipcktSourceRoute {sipckEntry 5}
Integer Read-Write

Specifies the behavior of IP routing with respect to (link-layer) source routing for this circuit. If default (1) then behavior depends on whether the port is token ring or not. If token ring then ARP requests are sent both as source routing explorers and as local frames; if not token ring then only local frames are sent. If sr (2) then ARP requests are sent as source routing explorers only. If no-sr (3) then ARP requests are sent as local frames only. If both (4), ARP requests are sent both as source routing explorers and as local frames. Default (1) should be adequate for most uses unless either source routing is being used on FDDI, or the ARP traffic is excessive on token ring.

10.3.22 sipNetToMedia Table

sipNetToMediaTable {ecs-1 22}
Not-Accessible

An extension to the standard ipNetToMedia table, containing other ARP cache attributes, specifically the current source route to the destination.

sipNetToMediaEntry {sipNetToMediaTable 1}
Not-Accessible

Each entry contains one IpAddress to “physical” address equivalence. Entries in this table exist even when there is no source route attached.

sipNetToMediaIfIndex {sipNetToMediaEntry 1}
Not-Accessible

The interface on which this entry’s equivalence is effective. The interface identified by a particular value of this index is the same interface as identified by the same value of ifIndex.

sipNetToMediaNetAddress {sipNetToMediaEntry 2}
Not-Accessible

The IP Address corresponding to the media-dependant “physical” address.

sipNetToMediaSourceRoute {sipNetToMediaEntry 3}
Octet String Read-Only

The source route to this destination.

10.3.23 Trunking Group

strunk {ecs-1 25}

This group deals with Cabletron’s trunking protocol.

strunkTable {strunk 1}
Not-Accessible

This table describes the trunking status and attributes of the interfaces that are configured for LAN trunking.

strunkEntry {strunkTable 1}
Not-Accessible

An entry exists for in this table for each physical interface (port) that is configured for trunking.

strunkIfIndex {strunkEntry 1}
Integer Read-Only

Identifies the physical interface (port) which is configured for LAN trunking. The value of this variable corresponds to ifIndex, as well as most of the other port identification values in related MIBs.

strunkState {strunkEntry 2}
Integer Read-Only

Indicates the trunking condition for this port. Values include:

- off (1) – this link has not been enabled for trunking
- closed (2) – this link has not yet received any trunking PDUs
- oneway (3) – incoming trunking PDUs do not indicate that the ATX's PDUs are being successfully received by the far end
- joined (4) – this link is actively participating in the trunk group
- perturbed (5) – this link is actively participating in the trunk group; however, the transmission of data packets has been temporarily stopped due to a change in membership of the trunk group
- helddown (6) – an error has been detected and this link is being held out of service until the error clears
- broken (7) – this link has been configured for trunking but the port is physically non-operational

strunkRemoteBridgeAddr {strunkEntry 3}
Octet String Read-Only

The MAC address portion of the bridge ID of the remote bridge.

strunkRemoteIp {strunkEntry 4}
IP Address Read-Only

The IP address of the remote bridge, as reported by that bridge for administrative purposes.

strunkLastError {strunkEntry 5}
Integer Read-Only

The reason for failure when the link is in closed, oneway, or helddown state. Values include:

- none (1) – no error; the trunking protocol may re-start with no error conditions when trunking is turned on for a port, or when the MIB variable that controls extra trunk groups is modified.
- in-bpdu (2) – a Spanning Tree BPDU was received, indicating that the connection is not point-to-point, or the far end does not have trunking turned on.
- multiple-bridges (3) – a different bridge has been connected at the far end and the trunking protocol will re-start.
- ack-lost (4) – the far end has detected a problem, and the trunking protocol will re-start.
- standby (5) – this trunk group is filled to capacity using other ports; this port is now a hot standby.
- too-many-groups (6) – the maximum number of groups an ATX can handle has been reached and a new group cannot be added. This port will not be used until the condition clears.
- no-ack (7) – this port has not received a valid trunking packet, and the trunking protocol will re-start.
- perturbed-threshold (8) – errors are preventing stabilization, and the trunking protocol will re-start.
- port-change (9) – this port is connected to another port on the same ATX and cannot be used until this condition clears.
- different-lan-type (10) – a different port has been connected at the far end, and the trunking protocol will re-start.
- multiple-lan-types (11) - several LAN types have been connected on the same ATX.

strunkLinkOrdinal {strunkEntry 6}
Integer Read-Only

The position of this link within the trunk group.

strunkLinkCount {strunkEntry 7}
Integer Read-Only

The number of links with the trunk group.

strunkLastChange {strunkEntry 8}
Integer Read-Only

The number of seconds since strunkState changed.

10.3.24 IP Multicast Route

ipm {ecs-1 26}

ipMRouteEnable {ipMRoute 1}
Integer Read-Write

The enabled status of IP Multicast routing on this router.

ipMRouteTable {ipMRoute 2}
Not-Accessible

The (conceptual) table containing multicast routing information for IP datagrams sent by particular sources to the IP multicast groups known to this router.

ipMRouteEntry {ipMRouteTable 1}
Not-Accessible

An entry (conceptual row) containing the multicast routing information for IP datagrams from a particular source and addressed to a particular IP multicast group address.

ipMRouteGroup {ipMRouteEntry 1}
Not-Accessible

The IP multicast group address for which this entry contains multicast routing information.

ipMRouteSource {ipMRouteEntry 2}
Not-Accessible

The source network address for which this entry contains multicast routing information.

ipMRouteSourceMask {ipMRouteEntry 3}
Not-Accessible

The network mask of the source address for which this entry contains multicast routing information.

ipMRouteRpfNeighbor {ipMRouteEntry 4}
IP Address Read-Only

The address of the RPF neighbor from which IP datagrams from this source to this multicast address are received.

ipMRouteInIfIndex {ipMRouteEntry 5}
Integer Read-Only

The value of IfIndex for the interface on which IP datagrams sent by this source to this multicast address are received.

ipMRouteOutList {ipMRouteEntry 6}
Integer Read-Only

The value of ipMRouteNextHopList which identifies the list of entries in the ipMRouteNextHopTable representing outgoing interfaces to which IP datagrams from this source to this multicast group address are forwarded.

ipMRouteUpTime {ipMRouteEntry 7}
TimeTicks Read-Only

The time since the multicast routing information represented by this entry was learned by the router.

ipMRouteExpiryTime {ipMRouteEntry 8}
TimeTicks Read-Write

The minimum amount of time remaining before this entry will be aged out.

ipMRoutePkts {ipMRouteEntry 9}
Counter Read-Only

The number of packets which this router has received from this source and addressed to this multicast group address.

ipMRouteRpfFails {ipMRouteEntry 10}
Counter Read-Only

The number of packets which this router has received from this source and addressed to this multicast group address, which were not received from the interface indicated by ipMRouteInIfIndex.

ipMRouteOctets {ipMRouteEntry 11}
Counter Read-Only

The number of octets contained in IP datagrams which were received from this source and addressed to this multicast group address, and which were forwarded by this router.

10.3.24.1 IPM Route Next Hop Table

ipMRouteNextHopTable {ipMRoute 3}
Not-Accessible

The (conceptual) table containing information on the outgoing interfaces and next hops for the routing IP multicast datagrams. Each entry is one of a list of outgoing interface and next hops for a particular source and multicast group address.

ipMRouteNextHopEntry {ipMRouteNextHopTable 1}
Not-Accessible

An entry (conceptual row) containing a particular outgoing interface and next hop for an IP multicast datagram.

ipMRouteNextHopList {ipMRouteNextHopEntry 1}
Not-Accessible

A value which uniquely identifies a list of outgoing interfaces and next hop addresses.

ipMRouteNextHopIndex {ipMRouteNextHopEntry 2}
Not-Accessible

A value which uniquely identifies a an outgoing interface and next hop address within a list of outgoing interfaces and next hop addresses.

ipMRouteNextHopState {ipMRouteNextHopEntry 3}
Integer Read-Only

An indication of whether the outgoing interface and next hop represented by this entry is currently being used to forward IP datagrams. The value “forwarding” (2) indicates it is currently being used; the value “pruned” (1) indicates it is not.

ipMRouteNextHopAddress {ipMRouteNextHopEntry 4}
IP Address Read-Only

The address of the next hop for this outgoing interface and next hop.

ipMRouteNextHopIfIndex {ipMRouteNextHopEntry 5}
Integer Read-Only

The IfIndex value of the interface for this outgoing interface and next hop.

ipMRouteNextHopUpTime {ipMRouteNextHopEntry 6}
TimeTicks Read-Only

The time since the multicast routing information represented by this entry was learned by the router.

ipMRouteNextHopExpiryTime {ipMRouteNextHopEntry 7}
TimeTicks Read-Only

The minimum time before this entry will be aged out.

10.3.24.2 IPM Route Interface Table

ipMRouteInterfaceTable {ipMRoute 4}
Not-Accessible

The (conceptual) table containing multicast routing information specific to interfaces.

ipMRouteInterfaceEntry {ipMRouteInterfaceTable 1}
Not-Accessible

An entry (conceptual row) containing the multicast routing information for a particular interface.

ipMRouteInterfaceIfIndex {ipMRouteInterfaceEntry 1}
Integer Not-Accessible

The ifIndex value of the interface for which this entry contains information.

ipMRouteInterfaceTtl {ipMRouteInterfaceEntry 2}
Integer Read-Write

The datagram TTL threshold for the interface. Any IP multicast datagrams with a TTL less than this threshold will not be forwarded out the interface. The default value of 0 means all multicast packets are forwarded out interface.

10.3.25 IGMP Interface

igmp {ecs-1 27}

igmpInterfaceTable {igmp 1}
Not-Accessible

The (conceptual) table listing the interfaces on which IGMP is enabled.

igmpInterfaceEntry {IgmpInterfaceTable 1}
Not-Accessible

An entry (conceptual row) representing an interface on which IGMP is enabled.

igmpInterfaceIfIndex {IgmpInterfaceEntry 1}
Integer Not-Accessible

The interface for which IGMP is enabled.

igmpInterfaceQueryInterval {igmpInterfaceEntry 2}
Integer Read-Write

The frequency at which IGMP Host-Query packets are transmitted on this interface.

igmpInterfaceStatus {igmpInterfaceEntry 3}
Integer Read-Write

The activation of a row enables IGMP on the interface. The destruction of a row disables IGMP on the interface.

igmpCacheTable {igmp 2}
Not-Accessible

The (conceptual) table listing the IP multicast groups for which there are members on a particular interface.

igmpCacheEntry {igmpCacheTable 1}
Not-Accessible

An entry (conceptual row) in the igmpCacheTable.

igmpCacheAddress {igmpCacheEntry 1}
IP Address Not-Accessible

The IP multicast group address for which this entry contains information.

igmpCacheIfIndex {igmpCacheEntry 2}
Integer Not-Accessible

The interface for which this entry contains information for an IP multicast group address.

igmpCacheSelf {igmpCacheEntry 3}
Integer Read-Write

An indication of whether the local system is a member of this group address on this interface.

igmpCacheLastReporter {igmpCacheEntry 4}
IP Address Read-Only

The IP address of the source of the last membership report received for this IP Multicast group address on this interface. If no membership report has been received, this object has the value 0.0.0.0.

igmpCacheUpTime {igmpCacheEntry 5}
TimeTicks Read-Only

The time since the system joined this group address, or zero if the system is not currently a member.

igmpCacheExpiryTime {igmpCacheEntry 6}
TimeTicks Read-Only

The minimum amount of time remaining before this entry will be aged out.

igmpCacheStatus {igmpCacheEntry 7}
Integer Read-Write

The status of this entry.

10.4 APPLETALK GROUP

This group contains information relevant to Cabletron's implementation of AppleTalk routing.

atext {sigma 6}

atextsystem {atext 1}

atextport {atext 2}

atextsysOperState {atextsystem 1}
Integer Read-Write

The operational state of AppleTalk routing. Values include off (1) and on (2).

atextportTable {atextport 1}
Not-Accessible

A list of AppleTalk ports for this entity.

atextportEntry {atextportTable 1}
Not-Accessible

The description of one of the AppleTalk ports on this entity.

atextportIndex {atextportEntry 1}
Integer Read-Only

A unique value for each AppleTalk port. Its value is between 1 and the total number of AppleTalk ports. The value for each port must remain constant at least from the re-initialization of the entity's network management system to the next re-initialization.

atextportNetStart {atextportEntry 2}
Octet String Read-Only

The first AppleTalk network address in the configured range for this port. This is a two octet DDP network address in network byte order.

atextportNetEnd {atextportEntry 3}
Octet String Read-Only

The last AppleTalk network address in the configured range for this port. This is a two octet DDP network address in network byte order.

atextportNetAddress {atextportEntry 4}
Octet String Read-Only

The AppleTalk network address configured for this port.

atextportZone {atextportEntry 5}
Octet String Read-Only

The default zone name configured for this AppleTalk port.

atextportzipTable {atextport 2}
Not-Accessible

The table of zone information configured for a port.

atextportzipEntry {atextportzipTable 1}
Not-Accessible

An entry of zone information for a particular zone and port combination.

atextportZonePort {atextportzipEntry 1}
Integer Read-Only

A unique value for each AppleTalk port.

atextportZoneIndex {atextportzipEntry 2}
Integer Read-Only

An integer that is unique to the ZoneName that is present in this entry. For any given zone name, every Entry that has an equal zone name will have the same ZoneIndex.

atextportZoneName {atextportzipEntry 3}
Octet String Read-Only

The ASCII zone name of this entry.

CHAPTER 11
TRAPS

The unit sends Trap PDUs to the NMS, using the pre-configured NMS IP address (see configNMSAddress). If no address has been pre-configured, then the unit sends the Traps to the source IP address of the last SNMP datagram received from an NMS. If no address has been pre-configured, and if no datagrams have been received since the unit was booted, then the unit uses the broadcast IP address.

The Trap PDUs are sent from source UDP port number 161, to destination UDP port number 162, which are the SNMP standard numbers reserved for Trap PDUs. The unit may be configured to send an additional copy of each Trap PDU to a user specified destination UDP port number (see sysTrapPort).

11.1 GENERIC TRAPS

SNMP defines seven different traps, which are described below.

- coldStart (0) – A unit has re-started, and the NMS's copy of the unit's configuration may be incorrect.
- warmStart (1) – Not used by the unit.
- linkDown (2) – A port has failed, and the unit's local management agent has disabled usage of the port. The "variable-bindings" portion of the trap contains the ifIndex of the port.
- linkUp(3) – A port has come back to life, and the unit's local management agent has re-enabled usage of the port. The "variable-bindings" portion of the trap contains the ifIndex of the port.
- authenticationFailure(4) – This trap is generated whenever the community name in a PDU does not match the corresponding password. All SetRequest PDUs must match the configAnyPass (refer to the description of configGetPass for SetRequest exceptions), GetRequest PDUs must match either the configGetPass or the configAnyPass. The GetRequest exception is for one of the debugging attributes; those PDUs must always provide the configAnyPass.

- egpNeighborLoss (5) – Not used by the unit.
- enterpriseSpecific (6) – The unit is reporting some interesting information, which is contained in the "variable-bindings" portion of the PDU. If the unit has been configured to require acknowledgments to its Trap PDUs (sysTrapAck), then the NMS must acknowledge the trap, generally by issuing a GetRequest for the significant variables involved in generating the trap.

  All enterpriseSpecific traps specify the value of sysObjectID.0 (i.e., the object identifier sysID) as the "enterprise", and zero as the "specific-trap". The "variable-bindings" may contain many pairs of variable names and their values.

  The unit can generate enterpriseSpecific traps in either an Acknowledge or a Non-Acknowledge mode. In the Non-Acknowledge mode, a single Trap PDU will be sent to the NMS. This PDU will contain sufficient information for the NMS to determine the cause of the Trap.

  In Acknowledge mode, the unit will re-send the PDU occasionally (see sysTrapTime and sysTrapRetry) until it receives an Acknowledgment from the NMS. The Acknowledgment is defined to be a GetRequest for the variables associated with the Trap's cause. Note that Acknowledge mode is only applicable for enterpriseSpecific traps; that is, only enterpriseSpecific traps may be re-sent.

  Many traps include additional variable bindings, besides the variable binding which initiated the trap to be generated. The additional variable bindings are not guaranteed to be present, and/or may be passed in separate Trap PDUs.

  The unit organizes its enterpriseSpecific Traps according to one of three different priorities. The highest priority Traps are sent, and re-sent, until they are acknowledged. After they are acknowledged (or immediately without waiting for acknowledgments, if sysTrapAck is False), the next highest priority Traps are sent, until they are acknowledged. Finally, the lowest priority Traps are sent, until they are acknowledged. The priority groups are described below.

11.2 PRIORITY 0 TRAPS

Priority 0 Traps are the highest priority enterpriseSpecific Traps, and are only sent if configAlarmDynamic is True. There is just a single element of this group, which is the trap that indicates that the unit has booted. That trap always contains configFatalErr, and is acknowledged by reading configFatalErr. In addition, that trap may contain any of the following for the modules which were configured to be used (refer to hwUseMod and rebootUseMod).

- hwDiagStatus – Included for any module which is not present or has failed power-up diagnostics.
- hwDiagCode – Included for any module which has a non-zero value to report.
- hwType – Included for any module which passed power-up diagnostics, but whose module type does not match the value defined for the module by hwDefType. Note that the unit's power-up processing will attempt to automatically redefine the configured hwDefType to match the module's actual type.
- hwPortType – Included for any module which passed power-up diagnostics, but whose port types do not match the port types defined for the module by hwDefPortType. Note that the unit's power-up processing will attempt to automatically redefine the configured hwPortType to match the module's actual port types.
- hwPortStatus – Included for any module which passed power-up diagnostics, but has one or more ports which have failed power-up diagnostics.

11.3 PRIORITY 1 TRAPS

Priority 1 Traps are the middle priority enterpriseSpecific Traps, and are only sent if configAlarmDynamic is True (except for swdisWriteStatus and swdisDesc). Each of these Traps is acknowledged by reading every variable within the Trap, unless otherwise indicated. The following variables may be reported.

- pppLinkQuality – Sent whenever the PPP port is enabled, and its link quality has changed (from bad to good, or from good to bad). When a PPP port is started, it is assumed to be good, until link quality measurements prove otherwise.
- fddimibSMTCFState – Sent whenever the FDDI port's CFM state has changed. The following parameters are also included.
- fddimibPORTRemoteMACIndicated – Sent in conjunction with fddimibSMTCFState (sent for both ports, if it is a DAS connection); fddimibPORTRemoteMACIndicated does not need to be acknowledged. The NMS should use this parameter to determine if the FDDI ring is twisted.
- fddimibMACUpstreamNbr – Sent whenever the FDDI port's upstream neighbor has changed. The following parameters are also included:

  fddimibMACOldUpstreamNbr
- fddimibMACOldUpstreamNbr – Sent in conjunction with fddimibMACUpstreamNbr; It does not need to be acknowledged.

- rptrBasHealthState – Sent whenever the operational state of the repeater has changed.
- hwDiagStatus – Sent whenever a module's diagnostics have completed (or a module's diagnostics were started but never completed), except for the initial diagnostics when the unit booted. The following parameters are also included, if the module's diagnostics have completed.
  - hwDiagCode
  - hwManufData
  - hwType – (only sent if the module passed diagnostics, and the module's type did not match hwDefType)
  - hwPortType – (only sent if the module passed diagnostics, and the port types of the module did not match hwDefPortType)
  - hwPortStatus – (only sent if the module passed diagnostics, and one or more of the module's ports failed diagnostics)
- hwDiagCode – Sent in conjunction with hwDiagStatus; hwDiagCode does not need to be acknowledged.
- hwManufData – Sent in conjunction with hwDiagStatus or hwTempOK; hwManufData does not need to be acknowledged.
- hwType – Sent in conjunction with hwDiagStatus; hwType does not need to be acknowledged.
- hwPortType – Sent in conjunction with hwDiagStatus; hwPortType does not need to be acknowledged.
- hwPortStatus – Sent in conjunction with hwDiagStatus; hwPortStatus does not need to be acknowledged.
- hwTempOK – Sent whenever the module's temperature transitions from too hot to okay, and vice versa. The following parameters are also included:
  - hwManufData
- hwFatalErr – Sent whenever a module dies unexpectedly. Since death of the ME or TURBO causes the unit to reboot, an hwFatalErr alarm for such a module will never occur.
- configPowerAc1 – Sent whenever the AC input to the unit's first power supply transitions from on to off, and vice versa.
- configPowerAc2 – Sent whenever the AC input to the unit's second power supply transitions from on to off, and vice versa.

- configPowerDc1 – Sent whenever the DC output of the unit's first power supply transitions from on to off, and vice versa.
- configPowerDc2 – Sent whenever the DC output of the unit's second power supply transitions from on to off, and vice versa.
- configPowerPresent1 – Sent whenever the presence of the unit's first power supply transitions from present to not present, and vice versa.
- configPowerPresent2 – Sent whenever the presence of the unit's second power supply transitions from present to not present, and vice versa.
- configStorageFailure – Sent if the unit's Configuration EEPROM has failed. The unit will not be able to reboot, and must be returned to the factory.
- configAuthenticationFailure – sent whenever the unit has sent the SNMP generic Authentication Failure Trap. The purpose of this trap is to provide a reliable (i.e., acknowledge-required) mechanism for an authentication failure. Therefore, multiple occurrences of this trap indicate either multiple authentication failures or a failure to receive the trap acknowledgment.
- ppeTxCongests – Sent whenever packets destined for the unit itself were discarded due to lack of buffer space.
- stRootID – Sent whenever the unit's Spanning Tree Root Bridge ID has changed. (This MIB variable is obsolete.)
- stRootCost – Sent whenever the unit's Spanning Tree cost to the Root Bridge has changed. (This MIB variable is obsolete.)
- stRootPort – Sent whenever the unit's Spanning Tree root port has changed. (This MIB variable is obsolete.)
- stTopChange – Sent whenever the unit has started or stopped the Spanning Tree topology change procedure.
- stActMaxAge – Sent whenever the unit's Spanning Tree Maximum Age has changed. (This MIB variable is obsolete.)
- stActHello – Sent whenever the unit's Spanning Tree Hello Time has changed. (This MIB variable is obsolete.)
- stActDelay – sent whenever the unit's Spanning Tree Forward Delay Time has changed. (This MIB variable is obsolete.)
- meshCostChange – Obsolete.
- meshSubnet – Obsolete.

- swdisWriteStatus – Sent when a bank of Flash EPROM has been erased. If swdisWriteStatus indicates success, then the unit is ready to be downloaded with the new software. The following parameters are also included:
  - swdisDesc
- swdisDesc – Sent in conjunction with swdisWriteStatus; swdisDesc does not need to be acknowledged.
- snmpsmtUpstreamRsp – Sent when the upstream neighbor of the requested FDDI device has been learned. The following parameters are also included:
  - snmpsmtUpstreamReq – Sent in conjunction with snmpsmtUpstreamRsp; This variable does not need to be acknowledged and is Obsolete.
  - snmpsmtUpstreamDescriptor – Sent in conjunction with snmpsmtUpstreamRsp; This variable does not need to be acknowledged and is Obsolete.
  - snmpsmtUpstreamState – Sent in conjunction with snmpsmtUpstreamRsp; This variable does not need to be acknowledged and is Obsolete.
- sifFunctions – Sent whenever the current functional state (active protocols) of the port has changed.
- sifCongested – Sent whenever outbound congestion control has been invoked for the port.
- sifState – Sent whenever the Spanning Tree state of the port has changed. This is obsolete.
- sifDesigCost – Sent whenever the cost to the Spanning Tree Root Bridge from the designated port of the LAN/WAN to which the port is attached has changed. This is obsolete.
- sifDesigRoot – Sent whenever the Root Bridge ID in received Spanning Tree Configuration BPDU's from the port has changed. This is obsolete.
- sifDesigBridge – Sent whenever the bridge ID of the Spanning Tree designated bridge of the LAN/WAN to which the port is attached has changed. This is obsolete.
- sifDesigPort – Sent whenever the port ID of the Spanning Tree designated port of the LAN/WAN to which the port is attached has changed.
- sifErrorsFlag – Sent whenever the number of hardware errors in received and transmitted packets has exceeded the port's limit. see sifThreshold and sifThresholdTime.

- sifRxQueues – Sent whenever the number of times that the port's receiver has stopped receiving packets due to buffer space shortages has exceeded the port's limit. See sifRxQueueThresh and sifRxQueueThreshTime.
- sifTxStormFlag – Sent whenever multicast storm protection has been invoked for the port.
- filterThreshFlag – Sent whenever usage of a port's combination filter has exceeded the filter's limits. See filterThreshold and filterThreshTime. The following parameters are also included:
  - filterLastSrc – Sent in conjunction with filterThreshFlag; This variable does not need to be acknowledged.
  - filterPortNo – Sent in conjunction with filterThreshFlag; This variable does not need to be acknowledged.
- sfddiShortAddressing – Sent whenever an FDDI packet is received (other than Claim/Beacon packets) with 16 bit MAC addresses.
- sfddiSmtConditionStatus – Sent whenever any of the indicated conditions has occurred. The trap is acknowledged by a SetRequest. Each bit that is set in the sfddiSmtConditionsStatus parameter within the SetRequest clears the corresponding alarm condition.
- sfddiSrfConditionsStatus – Sent whenever any of the indicated conditions has occurred. The trap is acknowledged by a SetRequest. Each bit that is set in the sfddiSrfConditionsStatus parameter within the SetRequest clears the corresponding alarm condition.
- sfddiSBFlag – Sent whenever the FDDI port's optical bypass becomes stuck, or un-stuck.
- sfddiOBSFuseBad – Sent whenever the fuse to the FDDI port's optical bypass becomes bad, or switched from bad to good.
- sfddiStationState – Sent whenever the FDDI port's Station State has changed.
- debugStringId – Sent whenever the unit has a debug text string to be displayed. The text strings are sent in a stream-like fashion, with no acknowledgments except for the last text string [when there are no more text strings, the unit re-sends the last text string, until it is acknowledged]. The following parameters are also included:
  - debugString – Sent in conjunction with debugStringId; This variable does not need to be acknowledged.

- IpbkOperation – Sent whenever the unit has finished a loop back test, or a loop back error has been detected. The following parameters are also included:
  - lpbkErrorNoReceives – Sent in conjunction with lpbkOperation; This variable does not need to be acknowledged.
  - lpbkErrorBadReceives – Sent in conjunction with lpbkOperation; This variable does not need to be acknowledged.
- swanLinkCost – Obsolete.
- swanMeshState – Obsolete.
- swanLinkSubnet – Obsolete.
- swanLinkBridge – Obsolete.
- swanLinkPort – Obsolete.
- srptrLinkStatusMask – Sent whenever the status of any of a repeaters's port has changed. This variable is obsolete.

11.4 PRIORITY 2 TRAPS

Priority 2 Traps are the lowest priority enterpriseSpecific Traps, and are only sent if both configAlarmAddresses and configAlarmDynamic are True. Each trap contains just the addrAlarmMAC parameter, and is sent if the learned information concerning that address table entry has changed. The trap is acknowledged by using the Addresses Configuration Group parameters to obtain the address table entry specified by addrAlarmMAC, and receiving a GetResponse with the addrMAC parameter.

APPENDIX A
PACKET TRANSLATION PROCEDURE

When you are setting up filters, if you choose to use the **Offset** field, it's helpful to know the header packet formats. Diagrams of the IP, UDP, and TCP headers are provided here for your reference.

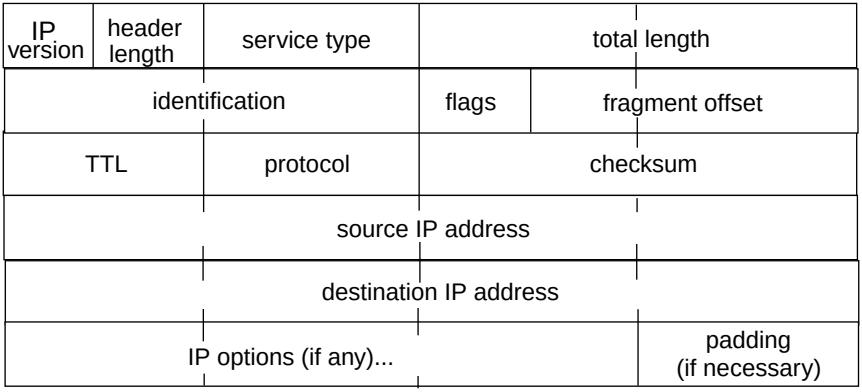


Figure A-1 IP Header

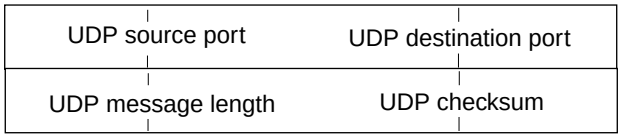


Figure A-2 UDP Header (After IP Header)

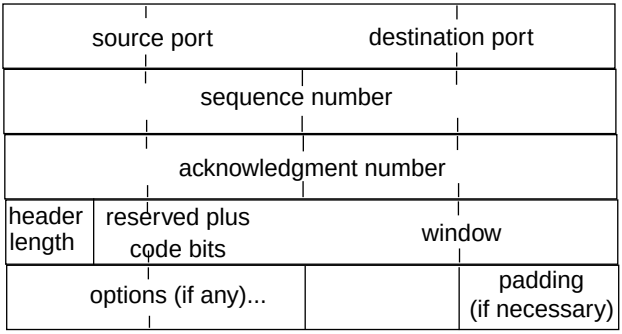


Figure A-3 TCP Header (After IP Header)